Looking Inside the Black-Box: Logic-based Explanations for Neural Networks*

Paper Review

Matthias Fulde

Goethe University Frankfurt am Main

31.7.2023

Abstract

A core problem of machine learning algorithms based on artificial neural networks is the lack of interpretability of these models. For many applications it is important to understand the reasoning behind the decisions made by an algorithm, but the inner workings of neural networks are generally opaque, leading to the notion of a black-box model. The hypothesis of the reviewed work is that human-understandable explanations of decision processes in neural networks, formulated as logic-based theories, can be obtained by mapping neural activations to semantic concepts that are used in these theories. The authors propose a procedure using a set of small networks to perform the mapping and an inductive reasoning framework to obtain the theories. It is empirically shown that for a synthetic classification task, the proposed procedure is able to generate logical theories, which are consistent both with existing theories for the task and with the predictions of the classification network, providing evidence that the induced theories describe the actual decision processes of the network on a human-understandable level. While being applicable to a broad class of machine learning algorithms based on neural networks in theory, it remains to be seen whether the approach can be used as a general tool for describing decision processes of neural networks for real world problems, since there remain significant challenges that have to be addressed in future work.

* João Ferreira, Manuel de Sousa Ribeiro, Ricardo Gonçalves, João Leite, 2022.

Contents

1 Introduction

This paper review is part of a seminar work. All included figures and tables are taken from the original paper [4].

1.1 Problem Statement

In the last decade, methods of artificial intelligence have seen an unprecedented rise in popularity. In particular, the development of increasingly powerful machine learning models that are based on artificial neural networks has enabled significant breakthroughs in various fields. However, while applications relying on artificial neural networks have become almost ubiquitous, a crucial challenge has arised, namely the lack of transparency and interpretability of these models.

For many applications in critical domains, such as medical diagnosis or autonomous driving, it is of utmost importance to understand the inner decision processes of the used algorithms. The ability to comprehend why a system arrives at a particular conclusion is essential for fostering trust, accountability, and ethical use of these technologies. This, however, contrasts with the inherent complexity and black-box nature of artificial neural networks.

Neural networks used for practical applications typically contain numerous layers with millions to billions of interconnected nodes, performing complex nonlinear computations on a purely numerical, subsymbolic level. This makes it extremely difficult for humans to understand how inputs are transformed into outputs. While several methods have been developed that shed some light on the inner workings of neural networks, they typically do not provide comprehensive explanations of the decision processes of a network that are understandable to end users, but merely provide some values for contribution to the network output for particular features in the input, or corresponding intermediate representations.

1.2 Approach

The authors of the reviewed paper propose a novel approach that allows to generate structured explanations for the decision processes of a neural network, formulated in logical languages, using human-understandable semantic concepts.

The idea is to first have domain experts annotate part of the input to a network of interest, with labels corresponding to concepts that are assumed to be constructive to the concepts that the network has been trained to predict, such that the predictions of the network could be explained in terms of these concepts. As an example, for a network trained to predict whether a bike is present in an image, the concept of wheel could be added under the assumption, that the decision whether an images contains a bike is based on whether wheels are present in the image.

In the second step, a set of small neural networks is trained to predict these added concepts from the neural activations of the main network. In particular, for each of the added concepts, a mapping network is trained to predict from the extracted activations whether the corresponding concept was present in the input that has generated these activations, thus relating the computations performed by the network of interest to the predefined concepts.

The final step is to use the predictions of the main network and the predictions of the mapping networks for a set of inputs to induce a logical theory that provides definitions of the predicted concepts of the main network in terms of the concepts predicted by the mapping networks. Since the definitions are formulated using human-understandable sematic concepts, they provide accessible explanations for the predictions of the main network.

2 Methods

The goal is to induce logic-based theories for predictions of neural networks based on concepts derived from the network's neural activations. Hence, the description of the procedure proposed in the reviewed work consists of the following steps: A specification of the class of networks taken into consideration, a description of the procedure to map network activations to concepts, and a description of how logical theories are induced from these concepts. These are provided in the following subsections. Figure 1 gives an overview over the proposed method.

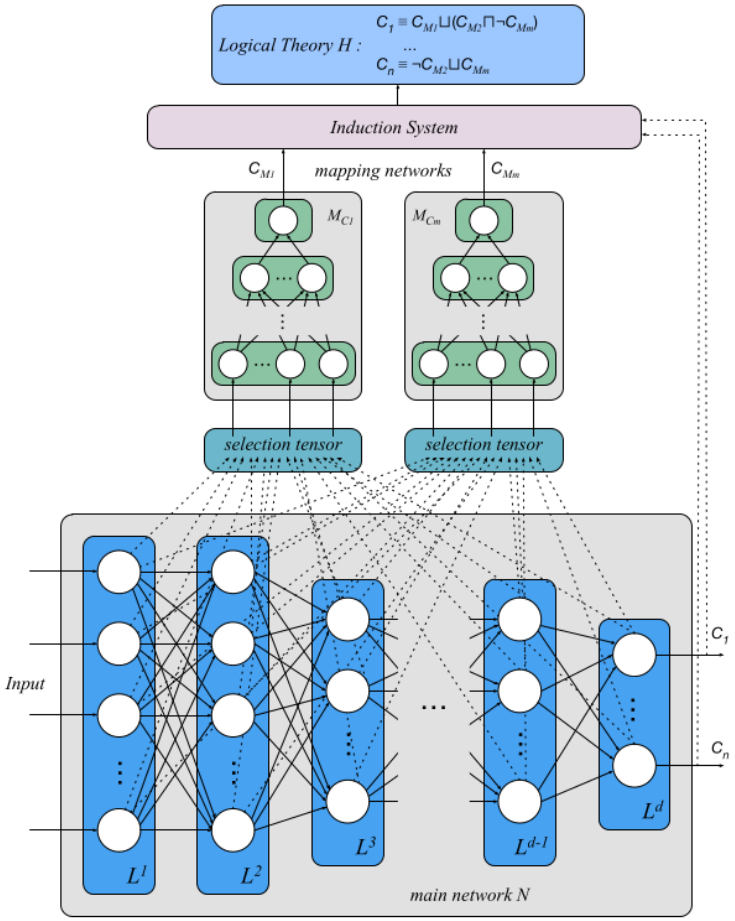

Figure 1: Architecture of the proposed method.

2.1 Neural Networks

The general class of neural networks taken into consideration by the authors is the class of feed-forward neural networks, which encompasses the majority of networks that are used in practice. The common characteristic of these networks is that the connections between the neurons form a directed acyclic graph. In these networks, information flows only in one direction, from input to output, without feedback loops. For computational reasons, neurons are almost always grouped in layers, such that the neurons within a layer are not connected to each other. Hence, the global structure of such a network can be described as a sequence of layers

$$N = (L^1, \ldots, L^d), \tag{1}$$

where each layer L describes a nonlinear mapping from some input tensor space $\mathbb{I}_L$ to some output tensor space $\mathbb{O}_L$, and the output of one layer is the input to the next. The elements of a layer's output tensor represent the neurons, and the values they take on for some particular input are called the activations of the neurons.

The inner structure of the layers is dependent on the architecture of the network, which is relevant for the implementation, but not for the conceptual description of the methods developed in the reviewed work. However, a common pattern is that each layer includes at least a linear projection that is followed by a nonlinear activation function, which is applied elementwise and produces the neural output [10].

Some particularities not mentioned explicitly by the authors when introducing their formalism to describe neural networks are the presence of special purpose layers often included for normalization or regularization, but these layers can be readily subsumed under the given definition, being regarded simply as components of the parametric layers of the network.

A classification network N is defined as a network whose output can be used to predict whether a particular concept C from a fixed set of predefined concepts $\mathcal{C}_N$ is present in the corresponding input to the network. More formally, let $\mathbb{I}_N = \mathbb{I}_{L^1}$ be the input space and $\mathbb{O}_N = \mathbb{O}_{L^d}$ be the output space of N. Furthermore, let $F_N : \mathbb{I}_N \to \mathbb{O}_N$ be the composition of the layers L in N. Then we can define a function

$$\text{concept}_N : \mathbb{O}_N \times \mathcal{C}_N \to \{0, 1\} \tag{2}$$

such that $\text{concept}_N(F_N(\mathbf{X}), C) = 1$ indicates that the concept $C \in \mathcal{C}_N$ is predicted by the network to be present in the input $\mathbf{X} \in \mathbb{I}_N$. Hence, the classification task with respect to a particular concept is a binary decision problem with a concept either being identified by the network or not.

2.2 Extracting Concepts from Neural Networks

The core idea of the proposed method is to have domain experts annotate input data to the main classification network N with a set of concepts $\mathcal{C}$, which are assumed to be relevant for the concepts $\mathcal{C}_N$ that the main network has been trained to predict. For each of the concepts $C \in \mathcal{C}$, a small mapping network M_C is trained to predict the concept from extracted activations of N. For a given training dataset, the outputs of the main network N and the mapping networks M_C are then given to an inductive reasoning

framework to induce theories for the concepts $\mathcal{C}_N$ in terms of the concepts $\mathcal{C}$, thereby providing interpretable explanations for the main network's predictions.

The mapping networks M_C are themselves feed-forward classification networks as described in the previous subsection. However, in contrast to the main network N, which is assumed to be trained to predict a larger set of concepts $\mathcal{C}_N$, each mapping network is associated only with a single concept $C \in \mathcal{C}$. That is, each mapping network takes a set of activations and predicts whether the associated concept is present in the input that generated these activations.

Since the input to the mapping networks are the activations of the main network N, the proposed method requires access to this network's intermediate activations. The neural activations of a network N are obtained by computing $F_N(\mathbf{X})$ for an input $\mathbf{X} \in \mathbb{I}_N$ and extracting the corresponding output tensors from each layer of the network. That is, the activations are given by an ordered list of tensors

$$(\mathbf{O}^1, \ldots, \mathbf{O}^d) \in \mathbb{O}_{L^1} \times \cdots \times \mathbb{O}_{L^d}. \tag{3}$$

Since the total number of activations of the main network N will in all but the most trivial cases be far too large to be processed by a mapping network in its entirety, a mechanism is required to select only a subset of the extracted activations. The decision which activation values to keep and which values to discard can be expressed in terms of tensors $\mathbf{S}^1, \ldots, \mathbf{S}^d$, which are binary masks with elements in $\{0, 1\}$, and have the same shape as the output tensors extracted from the network. The selection of activations for a particular mapping network can thus be described as

$$(\mathbf{S}^1 \odot \mathbf{O}^1, \ldots, \mathbf{S}^d \odot \mathbf{O}^d), \tag{4}$$

where $\odot$ denotes the Hadamard product, that is the pointwise product of its operands. Since each element in a mask $\mathbf{S}$ is either one or zero, the multiplication either retains or discards the corresponding activation from the output tensor $\mathbf{O}$.

A further restriction compared to the general classification network class is that the mapping networks are assumed to operate on vector input. Thus, instead of taking the ordered list of subsampled network activation tensors directly as their input, it is assumed that there exists a vectorization function that maps the tensors into a real vector space. The input for a particular mapping network is therefore given by

$$\mathrm{vec}(\mathbf{S}^1 \odot \mathbf{O}^1, \ldots, \mathbf{S}^d \odot \mathbf{O}^d). \tag{5}$$

However, while the introduced notation is convenient, it does not properly reflect how the input for the mapping networks is actually computed. Given the above notation, the size of the input to the mapping networks would still be equal to the number of neurons in the network that generated the activations, with some values just having been set to zero. Practically, the input size is equal to the number of activations retained by the mask, though.

An input of a mapping network M_C is thus given as a vector from $\mathbb{I}_M \subseteq \mathbb{R}^k$ where k is the number of selected activations. Since the network is used only to predict a single value, we may define the output space as $\mathbb{O}_M = [0, 1]$, which can be achieved by using the standard logistic function as the activation function for the final layer, in machine learning context known as the sigmoid function and defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{6}$$

In complete analogy to the definitions given for the main classification network, we can define a function $F_M : \mathbb{I}_M \rightarrow \mathbb{O}_M$ to compute the forward pass through a particular mapping network as the composition of the layers in the network. Moreover, we can define a function

$$\text{concept}_M : \mathbb{O}_M \rightarrow \{0, 1\} \tag{7}$$

such that $\text{concept}_M(F_M(\text{vec}(\mathbf{S}^1 \odot \mathbf{O}^1, \ldots, \mathbf{S}^d \odot \mathbf{O}^d))) = 1$ indicates that the concept associated with the mapping network is predicted from the activations in N, which can be achieved simply by checking whether the output value of the network is above or below 0.5. During training of the mapping network, the output of this function is compared to the labels that have been annotated to a subset of the overall training data for N by domain experts in a supervised fashion. During inference, the output is the label assigned to each data point in the larger dataset to be used by the inductive reasoning framework.

It is worth pointing out that the forward pass for each mapping network M_C includes a full forward pass through the main classification network N, necessary to generate the neural activations. This is true both for the inference and the training phase. Despite computing a full forward pass through N, the training is restricted to the parameters of the mapping network M_C, though. The parameters of the main classification network N are not altered in the process.

2.3 Inducing Logical Theories from Neural Networks

Given a main classification network N over a set of concepts $\mathcal{C}_N$, a collection of trained mapping networks M_C for concepts $\mathcal{C}$, relevant for the domain of N, the two remaining components of the method are a logical language that can be used to describe theories for $\mathcal{C}_N$ in terms of $\mathcal{C}$, and an induction framework which creates these theories from the concepts and individuals of the domain. In order to ensure a broad applicability of the method, only few requirements are formulated for the logical language and the corresponding induction framework.

In particular, a logical language $\mathcal{L}$ is only required to allow for the representation of concepts, relations between concepts, and knowledge specific to the individuals of the domain of interest. It is assumed that the used language $\mathcal{L}$ is defined over the union of the concept sets $\mathcal{C}_N$ and $\mathcal{C}$, and a set of constants

$$c_N = \{c_{\mathbf{X}} | \mathbf{X} \in \mathbb{I}_N\}, \tag{8}$$

which in logical language terms are the individuals, corresponding to the samples from the input domain of the main network N. The language $\mathcal{L}$ should include as basic formulas atoms of the form $C(c)$, where C is a concept and c an individual, asserting that c is an instance of C. In addition it is assumed that there is a semantic consequence relation $\models$ defined over the language $\mathcal{L}$. The semantic consequence relation allows the

inductive framework to induce an hypothesis, a set of formulas which is consistent with the main network's predictions and the concepts predicted from its activations.

The induction framework over $(\mathcal{L}, \models)$ consists of three components. The background knowledge BK is the set of atoms defined over the concepts $\mathcal{C}$, restricted to a subset $\mathcal{C}_\alpha \subseteq \mathcal{C}$ that only includes those concepts $C \in \mathcal{C}$ for which the accuracy of the corresponding mapping network M_C is higher than the threshold α. It contains the atoms $C(c_{\mathbf{X}})$ where for concept C and individual $c_{\mathbf{X}}$, corresponding to input $\mathbf{X}$, the mapping network M_C has predicted that C is present in $\mathbf{X}$. The two other components of the framework are sets of positive and negative examples from the input domain of N defined as:

$$\text{Pos} = \bigcup_{C \in \mathcal{C}_N} \{C(c_{\mathbf{X}}) | \mathbf{X} \in \mathbb{I}_N, \text{concept}_N(F_N(\mathbf{X}), C) = 1\} \tag{9}$$

$$\text{Neg} = \bigcup_{C \in \mathcal{C}_N} \{C(c_{\mathbf{X}}) | \mathbf{X} \in \mathbb{I}_N, \text{concept}_N(F_N(\mathbf{X}), C) = 0\} \tag{10}$$

The inductive framework then has the task to induce a hypothesis $H \subseteq \mathcal{L}$ such that for all $C(c) \in \text{Pos}$ holds that $BK \cup H \models C(c)$ and for all $C(c) \in \text{Neg}$ holds that $BK \cup H \not\models C(c)$.

3 Results

All experiments described in the reviewed work are classification tasks using a synthetic image dataset. After an initial proof of concept, additional results were obtained using different levels of abstraction in the concepts $\mathcal{C}$ defined to explain the main network's predictions, and using concepts $\mathcal{C}$ that are insufficient to describe the network's decision processes. It was tested how the accuracy of the mapping networks affects the quality of the induced theories, and an ablation study was conducted, replacing the output of the mapping networks with the labels added to the training data. In the following subsections, the dataset, the used networks, and the performed experiments are described in more detail.

3.1 Experimental Setting

The XTRAINS dataset [3] that was used throughout all experiments consists of 152×152 pixel color images, depicting sketched trains in front of colored backgrounds, such as those in figure 2. The trains vary in position, number and shape of wagons and wheels, and other geometric shapes. In addition, noise was added in form of missing pixels in the train's representation. The dataset is accompanied by an onthology represented using description logic, providing definitions such as

$$\text{Train} \equiv \exists \text{has}.(\text{Wagon} \sqcup \text{Locomotive}). \tag{11}$$

For all but the last experiment, three main networks $\mathcal{M}_A$, $\mathcal{M}_B$, and $\mathcal{M}_C$ were trained for binary classification, which possess different architectures, but are all convolutional neural networks, using convolutional, batch normalization [7], pooling, and dropout [6]

Figure 2: Example images from the XTRAINS dataset.

layers, with the final layers of the networks being fully-connected layers. These networks were trained to identify a single concept each, describing a particular type of train, using a subset of 25.000 training images from the XTRAINS dataset. The respective concepts TypeA, TypeB, and TypeC, are defined in terms of the presence or absence of particular geometric features. All networks achieved an accuracy of about 99% on a held out test set of 10.000 images.

The mapping networks that were trained for the performed experiments are shallow networks consisting only of an input and an output layer. These networks were trained on an annoteted subset of 800 images from the XTRAINS dataset and testet on a subset of 1000 images, except for the second to last experiment, where the size of the training datasets is varied between 50 and 1200 images. For inducing logical theories, only concepts have been taken into consideration for which the corresponding mapping network achieved at least $\alpha = 90\%$ accuracy.

All networks were trained using the Adam optimization algorithm and the binary cross-entropy loss function, which is defined as

$$L = -\frac{1}{N}\sum_{n=1}^{N}\left[t_n \ln(p_n) + (1 - t_n)\ln(1 - p_n)\right] \tag{12}$$

where N is the minibatch size, $t_n \in \{0, 1\}$ is the label, and $p_n \in (0, 1)$ is the normalized network prediction. A learning rate of 0.001 was used for training. Early stopping with a patience value of 20 for the mapping networks and 30 for the main networks was used to prevent overfitting.

As logical language, onthologies over description logics [1] were used to allow for direct comparsion with the dataset's associated onthology, and as induction system, the DL-Learner framework [9] was used, where an algorithm was chosen that is biased to minimize the induced theory. Across all experiments, for inducing the theories, a set of 3.000 images was used, and a set of 1 000 samples for testing.

For quantitative evaluation of the quality of the induced theories, two fidelity measures F_{Main} and $F_{XTrains}$ have been used. According to the authors of the paper, F_{Main} is computed as the ratio of samples where the classifications of the main network coincide with those obtained from the induced theory together with the knowledge obtained from the outputs of the mapping networks, and $F_{XTrains}$ is computed as the ratio of samples where the classifications of the XTRAINS labels (for the main network's output concepts $\mathcal{C}_N$) coincide with those obtained from the induced theory together with the knowledge obtained from the labels of the mapped concepts $\mathcal{C}$.

3.2 Inducing a Main Network's Theory

For each main network $\mathcal{M}_A$, $\mathcal{M}_B$, and $\mathcal{M}_C$, mapping networks were trained for the same set of 11 concepts $\mathcal{C}$, which are assumed to be constructive for the types of trains the main networks were trained to predict. As qualitative results, the following theories were induced:

$$\text{TypeA} \equiv \text{WarTrain} \sqcup \text{EmptyTrain} \tag{13}$$
$$\text{TypeB} \equiv (\text{FrightTrain} \sqcap \text{LongTrain}) \sqcup (\text{PassengerTrain} \sqcap \neg\text{EmptyTrain}) \tag{14}$$
$$\text{TypeC} \equiv \text{MixedTrain} \sqcup \text{RuralTrain} \tag{15}$$

The induced definitions of concepts TypeA and TypeC are equivalent to to those in the dataset's onthology, while the induced definition for TypeB is a subclass, meaning that any individual sample considered to be of TypeB by the induced theory would also be considered of this type according to the dataset's onthology.

The results reported above represent the mode of 20 repetitions of the experiment, where on average only 3 of the 11 concepts where used to define each of the types. The averaged fidelity scores obtained for the 20 runs are reported in the table in figure 3, showing that the induced theories are very much consistent both with the networks predictions and the XTRAINS labels.

	F_{Main}	$F_{XTrains}$
$\mathcal{M}_\mathsf{A}$	$99.72 \pm 0.18\%$	$99.92 \pm 0.35\%$
$\mathcal{M}_\mathsf{B}$	$98.71 \pm 0.31\%$	$99.83 \pm 0.76\%$
$\mathcal{M}_\mathsf{C}$	$99.33 \pm 0.32\%$	$99.52 \pm 1.52\%$

Figure 3: Fidelity scores for initial experiment.

3.3 Levels of Abstraction

The second experiment was a test how well the proposed method performs when faced with concepts $\mathcal{C}$ at different levels of abstraction. For the comparsion, a set of high-level concepts and a set of low-level concepts have been defined. In particular, the following concepts on the train level were used

$$\{\text{EmptyTrain}, \text{LongFreightTrain}, \tag{16}$$
$$\text{MixedTrain}, \text{PassengerTrain}, \text{RuralTrain}, \text{WarTrain}\} \tag{17}$$

as well as complex concepts on the wagon level, which, for the given classification task, were treated as atomic concepts. Examples for these concepts are

$$\{\exists\text{has.EmptyWagon}, \exists\text{has.FreightWagon}, \tag{18}$$
$$\exists\text{has.LongWagon}, \exists\text{has.}(\text{LongWagon} \sqcap \text{PassengerCar})\}. \tag{19}$$

As qualitative results, the induced theories from the train-level concepts were found to be logically equivalent to the definitions in the dataset's onthology. Regarding the wagon-level concepts, the induced definition for TypeB was a subclass of the onthology's definition, while the other definitions were neither subclasses nor superclasses to the corresponding definitions in the dataset's onthology.

The experiments were repeated 20 times again. The table in figure 4 summarizes the obtained average fidelity scores, which exhibit only slightly decreased values for the wagon-level concepts, while the 100% $F_{XTrains}$ scores obtained for the train-level experiments reflect the logical equivalence to the definitions in the onthology. The results suggest that the proposed method is able to induce high-quality theories for concepts with different levels of abstraction.

		F_{Main}	$F_{XTrains}$
Train-level	$\mathcal{M}_\mathsf{A}$	$99.76 \pm 0.19\%$	$100.00 \pm 0.0\%$
	$\mathcal{M}_\mathsf{B}$	$98.82 \pm 0.39\%$	$100.00 \pm 0.0\%$
	$\mathcal{M}_\mathsf{C}$	$99.46 \pm 0.20\%$	$100.00 \pm 0.0\%$
Wagon-level	$\mathcal{M}_\mathsf{A}$	$94.55 \pm 10.14\%$	$94.44 \pm 11.12\%$
	$\mathcal{M}_\mathsf{B}$	$97.50 \pm 0.52\%$	$96.73 \pm 1.18\%$
	$\mathcal{M}_\mathsf{C}$	$98.16 \pm 0.36\%$	$99.02 \pm 0.56\%$

Figure 4: Fidelity scores for different levels of abstraction.

3.4 Insufficient Concepts

Choosing concepts $\mathcal{C}$ that are insufficient to describe the concepts $\mathcal{C}_N$, which the main network was trained to predict, should result in lower quality theories. In order to rule out the possibility that the previously obtained results are attributed to spurious correlations in the data rather than accurate descriptions of the main network's decision process in terms of the concepts $\mathcal{C}$, another experiment was conducted, using 20 sets of 5 random concepts from the dataset's onthology to define $\mathcal{C}$.

As expected by the authors, the fidelity scores dropped significantly compared to the previous experiments. The table in figure 5 summarizes the results obtained in 20 runs of the experiment.

	F_{Main}	$F_{XTrains}$
$\mathcal{M}_\mathsf{A}$	$50.16 \pm 0.26\%$	$50.00 \pm 0.00\%$
$\mathcal{M}_\mathsf{B}$	$92.53 \pm 2.75\%$	$92.70 \pm 1.43\%$
$\mathcal{M}_\mathsf{C}$	$76.78 \pm 1.99\%$	$76.99 \pm 0.94\%$

Figure 5: Fidelity scores for insufficient concepts.

3.5 Theory Induction's Cost

Another experiment was performed to test whether the quality of the resulting theories mostly depends on the accuracy of the mapping networks. The same set of 11 concepts as in the initial experiment was used to induce theories while varying the amount of data used to train the mapping networks between 50 and 1200 samples. The amount of data that was used to induce the theories remained constant across all trials at 3000 samples.

A Pearson's correlation test on fidelity F_{Main} and the average accuracy of the mapping networks yielded a strong positive correlation of $r = 0.8161$ with a p-value of $p < 0.0001$. This indicates that when the mapping networks' accuracy increases, the quality of the induced theories increases as well.

3.6 Importance of the Mappings

The last experiment documented in the reviewed work was an ablation study where the mapping networks were removed from the procedure and instead of their predicted labels, fixed labels from the dataset were used for inducing logical theories. This test was included to assess whether the mapping networks are necessary to induce theories explaining the actual decision processes of a main network.

For this experiment, the XTRAINS dataset was augmented and a new classification task was constructed. In particular, the images in the dataset were augmented to include four traffic signals in varying top left, top right, bottom left, and bottom right positions in the image, such as in the images in figure 6. A traffic signal is said to be on if it contains any symbol in it. The images in the dataset are labelled with concepts

$$\{\text{On, TopLeftOn, TopRightOn, BottomLeftOn, BottomRightOn}\}. \tag{20}$$

The dataset is defined such that if one of the top signals is on, then also one of the bottom signals, and vice versa. A main network is tasked to predict the concept On, which is present in an image, if any of the signals is on. By construction of the dataset, for predicting the concept, it is sufficient for a network to either look at the top or bottom signals. For this experiment, 50 main networks with an accuracy of at least 90% were trained.

Figure 6: Augmented images including signals.

When using the dataset labels for inducing the theories over the main networks predictions, always the same theory was obtained, that is

$$\text{On} = \text{BottomLeftOn} \sqcup \text{BottomRightOn}. \tag{21}$$

Conducting the same experiment with labels from trained mapping networks, the obtained theories were much more diverse. 22% of the main networks learned to classify their outputs just by considering whether the two top signals were on while 42% looked at the two bottom signals. This suggests that the mapping networks are necessary in order to reflect the actual decision processes of the main network.

4 Discussion

While the authors of the reviewed work have shown empirically that the proposed method can work, results have been obtained only for a synthetic classification task, raising the question how well the method generalizes to real world problems.

In particular, the synthetic XTRAINS dataset used in all experiments does not reflect the properties of naturalistic image data used in practice, despite being constructed to have this appearance. The data that is actually relevant for the classification task lies on a comparatively low dimensional latent manifold, and the features are almost discete, such that finding a separable space is rather trivial. This is documented by the unrealisticly high accuracies that were obtained in the performed experiments. Given that accuracy appears to be a crucial factor for the quality of the induced theories, it is doubtful whether comparable results can be obtained for more complex problems, where accuracy is typically significantly lower. The lack of results under more realistic assumptions makes it hard to assess the applicability of the method in practice.

In addition, there are two particular issues to consider that could restrict the scalability of the method. The problem of manually labelling parts of the data and the problem of selecting subsets of neural activations to serve as input for the mapping networks. These problems are discussed in more detail in the two following subsections.

4.1 Labelling Problem

The authors of the reviewed work conclude that the main cost of their method is the labelling of the data to train the mapping networks M_C. Assuming that this was true, this cost alone might still be too high to make the method applicable to many, if not most problems encountered in practice.

There are two main difficulties with labelling the data. First of all, even if we assume that for each concept $C \in \mathcal{C}$, only a fairly limited amount of labelled training data is required, the number of concepts relevant to explain all concepts $\mathcal{C}_N$, that the main network N has been trained to recognize, can still be too large to allow for manual annotation. In addition, finding suitable concepts $\mathcal{C}$ for explaining the concepts in $\mathcal{C}_N$ can itself be a non-trivial and prohibitively time consuming task, even for experts of the respective domains.

We observe that state of the art neural network models are typically trained on very large amounts of high dimensional, very diverse and often also multimodal data, which enables them to recognize thousands or even millions of different concepts. Moreover, many advanced models exhibit significant zero-shot capabilities, allowing them to predict

concepts during inference that have not been part of the original training data at all. For these kind of models, manually labelling enough data to allow for the induction of high quality theories, for all of the concepts that the network might predict, seems to be hopeless.

Hence, unless some form of automated labelling can be applied, the proposed method will likely be applicable only in very restricted settings, like particular cases of medical diagnosis or specialized applications in manufacturing environments, where the number of relevant concepts is limited and sufficient domain knowledge is available.

4.2 Selection Problem

A major issue with the approach introduced in the reviewed work, which is not addressed by the authors, is the problem of selecting the set of neural activations from the main network N, that should be used as input to a particular mapping network M_C. While the authors acknowledge that taking the entire set of activations as input is computationally intractable, due to the large number of neurons in networks used in practice, they do not provide a strategy for how to choose the network activations to predict a predefined concept from. This is a severe limitation, since an unfavorable choice of the activation values will likely degrade the performance of the mapping networks below an acceptable threshold, removing the corresponding concepts from the background knowledge that is used for inducing the theories over the predictions of N.

As an example, consider a deep neural network trained for image classification, and a particular input image, depicting an airplane, which shall be one of the concepts $\mathcal{C}_N$ that the network was trained to recognize. Now, a domain expert might have come to the conclusion that 'wing' is a concept of interest. Hence a mapping network should be trained to predict this concept from the activations of the main network, and we're tasked to define a selection of network activations that is used as input to that mapping network.

A well-known property of deep neural networks and one of the foremost reasons for their ability to learn difficult tasks in an end-to-end fashion, is that they learn concepts in a hierarchical manner. That is, while early layers in the network recognize simple concepts, subsequent layers recognize increasingly abstract concepts that are composed of the simpler concepts recognized by previous layers. Analogous to the processes performed by the ventral stream of the visual cortex, as an instantiation of the property described above, deep neural networks trained for vision tasks encode a hierarchy of visual concepts across their layers, where the first few layers learn to recognize graphical primitives like lines, curves, and simple patterns, while deeper layers learn increasingly complex shapes [8].

Taking this into consideration, choosing a subset of activations from the first layer of the classification network to predict the concept 'wing' will likely be a suboptimal choice. While the complex concept 'wing' is correlated with the presence of certain graphical primitives, in the general case, there will be no clear causal relationship between the activations of the neurons in the first layer and this particular concept, because the concept itself is encoded in deeper layers and the graphical primitives it is composed of could also belong to different complex objects, like for example a conference table or a canopy. Hence, it can be expected that in general, the accuracy of the mapping network will not be sufficient for the concept to be included in the theories that are meant to explain why the network recognizes an airplane or not.

Unfortunately, while constructing a counterexample like this is relatively easy, the inverse problem of finding the areas of a network that encode a particular concept is much harder, and might be of the same order as the problem that the proposed method is meant to solve.

In particular, the search space increases when the size of the main network increases and the predefined input size to the mapping networks decreases. While neural networks used for complex real world applications will grow to large sizes, it is desirable to restrict the input size to the mapping networks for computational reasons, thus requiring more precise estimates of where in a main network the concepts of interest are encoded. Given the potentially large discrepancy between the total amount of neurons in the network of interest and the limited size of the selected subsets, such estimates likely cannot be obtained just by utilizing the general knowledge about hierarchical representations in deep neural networks, but demand a thorough investigation of the network whose decision processes we aim to explain.

To make matters worse, most tools developed to relate neural activations to particular concepts cannot be used in this case, because the concepts $\mathcal{C}$ that are predicted by the mapping networks are precisely the concepts that are not included in the main network's output. Hence, neither occlusion experiments [14] nor gradient based methods [5, 11, 12] developed to relate neural activations to concepts learnt by the main network are applicable, since there is no known outcome for these concepts that can be traced back to the respective areas of interest.

Now, it seems reasonable to assume that more powerful mapping networks could compensate to some degree for suboptimal choices of neural activations, being able to find patterns unique to particular concepts even from regions of the main network that are only weakly correlated to the respective concepts. That means, there is probably a tradeoff between the computational complexity required for the training of the mapping networks and the required accuracy of the selections. However, this could easily push the computational cost beyond the threshold that is deemed acceptable for a particular application. Moreover, this approach could also introduce false assumptions about the actual computations in the main network.

One way to reduce the search space could be to take into account the receptive fields of the neurons in the main network. The receptive field of a neuron is the area of the input that affects the neuron's activation, or, in other words, the part of the input that the neuron can see. In many network architectures used in practice, the size of the receptive field is fixed and positively correlated with the depth of the layer where a neuron resides. Neurons in early layers can only see a small portion of the entire input, while neurons in deeper layers can see larger parts. While some of the most advanced network architectures use receptive fields that are adaptive to the input data [2, 13], making it more difficult to determine the area of the input that affects particular neurons, in many settings it will be possible to discard neurons from the selection in an automated way, given that the input is annotated such that the location of a concept is known. This, however, will again increase the cost of labelling the data, and might not always be possible.

To summarize, while the authors provided empirical results relating the mapping network's accuracy to the quality of the induced theories, it would have been equally important to investigate the relationship between the selection of neural activations and the accuracy of the mapping networks.

4.3 Future Work

In order for the proposed method to gain any traction, it is absolutely mandatory to test it in a realistic setting. Even for a proof of concept paper, where initial tests on synthetic data can be reported, it would have been highly desired that at least one test under realistic assumptions was included, using a common benchmark dataset. This is a shortcoming that must be made up for in future work.

Besides that, the concerns raised in the previous subsections should be addressed, which could also involve some further development of the method that was suggested by the authors themselves, namely to change the method such as to induce probabilistic theories based on the accuracies of the mapping networks.

This approach could also be extended for automated labelling of data with concepts to be learnt by the mapping networks, which as seen above, will likely be a requirement for many real world applications. The benefit would be that probabilistic models of the main networks decision processes would also allow to explicitly handle the inevitable error that is introduced by using other neural networks to perform the labelling at the initial stage.

References

[1] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. The description logic handbook: Theory, implementation, and applications. *Kybernetes*, 32, 2003.

[2] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017.

[3] Manuel de Sousa Ribeiro, Ludwig Krippahl, and João Leite. Explainable abstract trains dataset. *ArXiv*, abs/2012.12115, 2020.

[4] João Ferreira, Manuel de Sousa Ribeiro, Ricardo Gonçalves, and João Leite. Looking inside the black-box: Logic-based explanations for neural networks. *Proceedings of the Nineteenth International Conference on Principles of Knowledge Representation and Reasoning*, 2022.

[5] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2013.

[6] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *ArXiv*, abs/1207.0580, 2012.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[8] E. Kobatake and K. Tanaka. Neuronal selectivities to complex object features in the ventral visual pathway of the macaque cerebral cortex. *Journal of Neurophysiology*, 71(3):856–867, 1994. PMID: 8201425.

[9] Jens Lehmann. Dl-learner: Learning concepts in description logics. *J. Mach. Learn. Res.*, 10:2639–2642, 2009.

[10] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. In *Psychological review*, volume 65, page 386, 1958.

[11] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

[12] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014.

[13] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4784–4793, 2022.

[14] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, 2013.