Aligning Artificial Neural Networks and Ontologies towards Explainable AI

Manuel de Sousa Ribeiro, João Leite
NOVA LINCS, School of Science and Technology, NOVA University Lisbon, Portugal
mad.ribeiro@campus.fct.unl.pt, jleite@fct.unl.pt

Abstract

Neural networks have been the key to solve a variety of different problems. However, neural network models are still regarded as black boxes, since they do not provide any human-interpretable evidence as to why they output a certain result. We address this issue by leveraging on ontologies and building small classifiers that map a neural network model's internal state to concepts from an ontology, enabling the generation of symbolic justifications for the output of neural network models. Using an image classification problem as testing ground, we discuss how to map the internal state of a neural network to the concepts of an ontology, examine whether the results obtained by the established mappings match our understanding of the mapped concepts, and analyze the justifications obtained through this method.

1 Introduction

In this paper, we investigate whether it is possible to relate the activations of a neural network with human-defined concepts from an ontology, with the prospect of finding symbolic justifications for the output of that neural network through automated reasoning. Our results suggest that this is possible for concepts that are relevant to the task of the neural network with little overhead, requiring only a small amount of labeled data and the training of small classifiers.

In the last few years, artificial neural network-based machine learning methods have allowed the field of Artificial Intelligence to successfully address multiple complex tasks, such as image (Sultana, Sufian, and Dutta 2019), video (Karpathy et al. 2014) and speech recognition (Graves, Mohamed, and Hinton 2013), translation (Huang et al. 2013) and text analysis (Lai et al. 2015), helping with drug design and discovery (Segler et al. 2018), and reconstruction of brain circuits (Lee et al. 2019), to name a few. Some of the systems developed by these methods outperformed humans in tasks like facial recognition (Schroff, Kalenichenko, and Philbin 2015) or playing strategic games (Berner et al. 2019; Silver et al. 2017). The successes achieved by neural network-based methods rendered them a widespread solution with innumerous applications, ranging from intelligent cameras with object recognition (Redmon et al. 2016) to predicting river discharges and floods (Kratzert et al. 2019).

As these systems begin to act autonomously and start being responsible for decisions previously taken by humans, like self-driving cars (Bojarski et al. 2016) and unmanned aerial vehicles (Radovic, Adarkwa, and Wang 2017), or start being applied in sensitive domains, such as medical diagnosis and treatment recommendation (Topol 2019), the need for humans to understand their reasoning becomes evident. Explanations of how a system behaves and why it outputs a certain result are known to allow users to build trust in a system and its results (Pieters 2011), to increase the chances of users taking action based on a system's output (Ye and Johnson 1995), and for a better assessment of when a system is right or wrong (Biran and McKeown 2017; Gkatzia, Lemon, and Rieser 2016).

However, since neural networks generally only use representations based on high-dimensional Euclidean space, i.e., real-valued vectors, matrices, etc., which possess no obvious associated declarative meaning (Hitzler et al. 2020), there is no direct human interpretable indication of why a specific output was given. Hence, to justify the output of a neural network, a language containing human-understandable concepts and meaningful relations between those concepts is needed, allowing for a comprehensible description of the reasoning that led the neural network to attain its output. The level of abstraction and detail of the justifications is dependent on the defined language. For example, when faced with a picture of a representation of a train, c.f. Figure 1, we might expect a justification as to why this train is of some particular type, e.g., a passenger train, to refer to the particular lines, squares, and circles that can be identified as constituting the train representation, or we might expect a justification to be based on high-level concepts such as the existence of passenger wagons or freight wagons. For each domain of interest, even though there are relationships between the different levels of abstraction, we, humans, are typically interested in a given particular level, where we can conveniently justify ourselves to other humans.

The field of knowledge representation and reasoning provides many formalisms that allow the description of domains of interest, such as ontologies. An ontology is the conceptualization of a domain, through the use of concepts and axioms, and is usually specified using a logic-based language with a precise semantics. Among the most important formalisms in which to describe ontologies are Description

Figure 1: Sample images of trains' representations.

Logics (Baader et al. 2003), on which the W3C standard OWL is formally based. Description Logics typically consist of a decidable fragments of First-Order Logic. Ontologies in Description Logics are typically composed of a TBox and an ABox. The TBox contains knowledge in the form of a terminology and is made of axioms describing how the domain's concepts relate to each other. The ABox contains assertional knowledge and is made of axioms that describe knowledge specific to the individuals of the domain of discourse. While the knowledge in the TBox is usually thought to be unchanging, the knowledge in the ABox is usually thought to be contingent. Initiatives like the Semantic Web (Berners-Lee, Hendler, and Lassila 2001), and Linked Open Data fostered the availability of many ontologies of different domains, often working as a network where the concepts of an ontology may be related to concepts of other ontologies.[1]

An ontology can therefore be used to define the necessary language (concepts and relations), at the appropriate level of abstraction, to adequately convey the justifications for the output of a given neural network. However, for a neural network's internal representations to be presented in a meaningful and human-understandable way, we need to establish some mapping to the concepts existing in the ontology. To this end, we found inspiration in the research conducted in the field of neuroscience, where ensembles of neurons and how they respond to stimuli have been investigated to comprehend what information they encode (Hassabis et al. 2009). By mapping the stimulus to the response, neuroscientists were able to understand the function and information encoded in these neuronal ensembles (Quiroga et al. 2005).

We propose to establish mappings from the values of the activations produced by the neurons of a neural network to concepts from a chosen ontology. Then, when feeding input to the neural network, we can observe, through these mappings, whether the corresponding concepts were identified in the generated activations, acquiring knowledge about the input's characteristics. Using logic based reasoning methods, together with the ontology and the observations made regarding each mapped concept, we can create a justification for the neural network's output. The justifications would be minimal sets of axioms from the ontology that, together with the observations, entail the output of the neural network.

One might wonder how much do the justifications produced by this method depend on the ontology, and whether it can be said that they represent an explanation of the neural network's internal output generation process. It should be pointed out that we, humans, can often reason in the most varied domains, even without knowing any specific language to describe it. Once we are taught concepts and their relationships in a given domain, we are usually able to map our experiences and intuitions into them, allowing us to produce meaningful explanations of our internal mental processes. For instance, we might be able to understand a given geometric phenomenon without knowing anything about coordinate systems, and once we learn the Cartesian coordinate system, we can use it to explain that phenomenon. Nonetheless, we could have learned the polar coordinate system instead, and still use it to describe the same phenomenon. While both explanations would certainly be different, and likely none of them would correspond literally to our internal understanding of the phenomenon, both would be meaningful and plausible, which is often enough.

In this paper, we explore the generation of justifications for artificial neural networks through the use of mappings between their neuron's activations and concepts of an ontology, which we then leverage with sound reasoning methods.

In the remaining of this paper, we analyze the intricacies involved with the process of mapping human-defined concepts from trained neural networks and examine the kind of results that might be expected by this approach; we compare the concepts resulting from this method and examine how they resemble our understanding of those concepts; we present a procedure for pinpointing which neurons should be used to map a given ontology's concept; and we discuss how to obtain justifications for the output of trained neural networks based on the extracted concepts.

2 The Main Networks

To illustrate the proposed method, we test it in a setting where images with simplified representations of trains, such as the ones shown in Figure 1, are classified based on their visual features. The images used in this paper's dataset (de Sousa Ribeiro, Krippahl, and Leite 2020) were inspired by those developed by J. Larson and R. S. Michalski in (Larson and Michalski 1977) and use fragments of images from (Olmos and Kingdom 2004) as background. The trains' representations are amply diverse, varying in the number, size, and shape of the trains' wagons and wheels, and in the quantity, size and relative position of the geometric shapes inside each wagon, but also in the distance between each wagon, the thickness of the wagons' walls, the height of the trains' couplers, etc. Noise was explicitly introduced in the form of missing pixels from the trains' representations.

Each image in the dataset was labeled as being, or not, of three different types of trains:

- Type A – trains having either a wagon with at least a circle inside and a wagon with two walls in each side, or no wagons with geometric figures inside them;
- Type B – trains having a long wagon or two wagons with at least a circle inside, or trains having at least two long wagons, or three wagons, with at least two of which with a geometric figure inside that is not a circle;
- Type C – trains having a wagon with no geometric figure inside, and either a wagon with a circle inside and a wagon with a geometric figure inside that is not a circle, or no long wagons and a wagon with a geometric figure inside.

[1] We assume basic familiarity with Description Logics.

$$\exists \text{has.EmptyWagon} \sqcap \exists \text{has.}(\text{PassengerCar} \sqcup \text{FreightWagon}) \sqcap \neg\exists \text{has.LongWagon} \sqsubseteq \text{RuralTrain}$$
$$\text{Train} \equiv \exists \text{has.}(\text{Wagon} \sqcup \text{Locomotive}) \qquad \exists \text{has.FreightWagon} \sqcap \exists \text{has.PassengerCar} \sqcap \exists \text{has.EmptyWagon} \sqsubseteq \text{MixedTrain}$$
$$\text{TypeA} \equiv \text{WarTrain} \sqcup \text{EmptyTrain} \qquad \exists \text{has.}(\text{PassengerCar} \sqcap \text{LongWagon}) \sqcup (\geq 2\ \text{has.PassengerCar}) \sqsubseteq \text{PassengerTrain}$$
$$\text{TypeB} \equiv \text{PassengerTrain} \sqcup \text{LongFreightTrain} \qquad \exists \text{has.ReinforcedCar} \sqcap \exists \text{has.PassengerCar} \sqsubseteq \text{WarTrain}$$
$$\text{TypeC} \equiv \text{RuralTrain} \sqcup \text{MixedTrain} \qquad (\geq 2\ \text{has.LongWagon}) \sqcup (\geq 3\ \text{has.Wagon}) \sqsubseteq \text{LongTrain}$$
$$\text{LongFreightTrain} \equiv \text{LongTrain} \sqcap \text{FreightTrain} \qquad (\geq 2\ \text{has.FreightWagon}) \sqsubseteq \text{FreightTrain}$$
$$\text{EmptyTrain} \equiv \forall \text{has.}(\text{EmptyWagon} \sqcup \text{Locomotive}) \sqcap \exists \text{has.EmptyWagon}$$

Figure 2: A subset of the ontology's axioms, describing how the trains' representations are classified.

Throughout this paper we report on several experiments performed on three different convolutional neural network models – referred to as $\mathsf{NN_A}$, $\mathsf{NN_B}$, and $\mathsf{NN_C}$ – trained to identify trains of each corresponding type. Each neural network was trained with a balanced dataset of $25\ 000$ images and achieves an accuracy of about 99% on a balanced test set of $10\ 000$ images. All three neural networks possess a different architecture, although each possesses at least a set of convolutional, batch normalization, pooling, and dropout layers followed by a set of fully connected and batch normalization layers, with a single output neuron at the end.

3 The Ontology

Starting with a previously trained neural network – in our case, three – whose output we are interested in justifying, we now need an ontology that defines the adequate language necessary to build the justifications. A publicly available ontology might be used, if it encompasses the domain of the task of the neural network and contains concepts analogous to those output by the neural network, or an ontology might be designed purposely to define the language to justify the output of the neural network. The more comprehensive the ontology, the more detailed the justifications.

To illustrate our proposal in the case of justifying the output of the neural networks $\mathsf{NN_A}$, $\mathsf{NN_B}$, and $\mathsf{NN_C}$, we designed an ontology using high-level concepts that allow for a simple and intuitive description of the trains' components – the kind that humans would know about and expect – such as, for example, PassengerCar, represented by a wagon of any shape and size containing, at least, a circle inside; or FreightWagon, represented by a wagon of any shape and size containing inside geometric figures that are not circles. A subset of this ontology is shown in Figure 2, illustrating how other concepts can be further introduced and interrelated, such as, for example, FreightTrain as encompassing those with at least two FreightWagons, and RuralTrains as including those having an empty wagon, either a passenger car or a freight wagon, and no long wagons.

Note that even though the dataset was labeled with the concepts used in the ontology, no other label apart from TypeA, TypeB and TypeC, nor the knowledge encoded in the ontology was used in any way in the process of developing and training the neural networks $\mathsf{NN_A}$, $\mathsf{NN_B}$, and $\mathsf{NN_C}$.

4 The Mapping Networks

We now focus on the central part of our proposal, namely to relate the information encoded in a neural network – dubbed *main network* – with the concepts in the ontology. We will do so by approximating each unknown mapping from the activations of a neural network to a single ontological concept through the use of another neural network – dubbed a *mapping network*. Mapping networks are trained to output whether a given activation pattern – a set of neurons' activations – from the main network represents an individual belonging to a given concept. We refer to the act of training a mapping network to predict a concept as *concept extraction*. It is important to note that mapping networks might be seen as an independent tool given that their use does not modify or require the retraining of the main network. They go beyond the classifiers in (Alain and Bengio 2017) and (Kim et al. 2018) that are limited to linear combinations of the neuron's activations, which is in general insufficient to extract human-defined concepts such as those in an ontology.

To test the use of mapping networks to extract concepts from neural network models, multiple experiments were performed. Each mapping network took as input the activations fed to and produced in the dense part of its main network, with the exception of the mapping networks developed in Section 4.4, where we discuss how to choose the set of input features. All activation values were extracted from the batch normalization layers of the main networks – $\mathsf{NN_A}$, $\mathsf{NN_B}$, or $\mathsf{NN_C}$. The results of each experiment are averaged over 20 repetitions, using different balanced sets of samples for training, validating, and testing purposes. All neural networks were trained using the optimization algorithm Adam (Kingma and Ba 2015), with a learning rate of 0.001, the binary cross entropy as loss function, and early stopping with a patience value of 15 for mapping networks and 30 for convolutional neural networks. Two different mapping network architectures were considered, one where the mapping network is composed by a single neuron, and another where the mapping network is composed by three layers of fully connected neurons, containing 10, 5 and 1 neurons respectively. All mapping networks use the ReLU activation function (Nair and Hinton 2010) in their hidden layers, and the sigmoid activation function in their output layers.

4.1 The Relevant Concepts

Our main hypothesis is that if human-defined ontological concepts are relevant to the task of a trained neural network, then we should be able to relate them to the representations encoded in the model of that network. For instance, if a neural network were trained to identify *mixed trains*, then we should be able to relate the representations encoded in the network's model with concepts like PassengerCar and FreightWagon, given that they are generally used to define MixedTrains. In general, we expect to be able to extract from

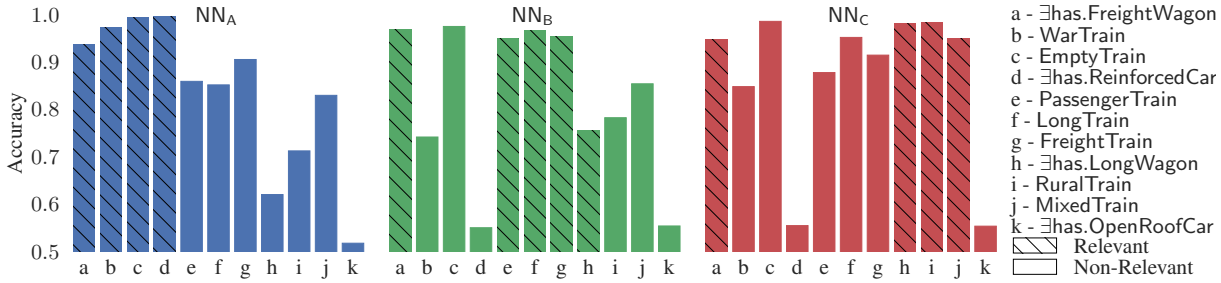

Figure 3: Accuracy of each mapping network trained to extract concepts based on the activations of $\mathsf{NN_A}$, $\mathsf{NN_B}$, and $\mathsf{NN_C}$.

a neural network concepts that are relevant to its task. Informally, a concept is relevant to another concept if there are circumstances where knowledge relative to the former allows us to infer knowledge about the latter. Formally, concept C_1 is relevant to concept C_2, with respect to an ontology TBox $\mathcal{T}$, if there exists an ABox $\mathcal{A}$, composed only by atomic assertions or their negations, such that (where x is a fresh individual):

- $(\mathcal{T}, \mathcal{A}) \not\models C_2(x)$
- $(\mathcal{T}, \mathcal{A} \cup \{C_1(x)\}) \models C_2(x)$
- $(\mathcal{T}, \mathcal{A} \cup \{C_1(x)\}) \not\models \bot$

A concept is said to be *relevant* to a neural network, with respect to a given ontology, if it is relevant to a concept that is analogous to an output of that network. For example, if a neural network is trained to identify *long freight trains*, then LongFreightTrain would be the analogous concept, and $(\geq 2\ \mathsf{has.FreightWagon})$ a relevant concept to the network.

According to our hypothesis, we expect that we should generally achieve better results when extracting relevant concepts than non-relevant concepts from a given main network. To test our hypothesis we built mapping networks to extract various (11) concepts defined in the ontology shown in Figure 2 from all three main networks. Each mapping network was trained with a set of 800 samples, selected using a validation set of 200 samples, and tested using a set of 1 000 samples, each labeled with the concepts being extracted.

In Figure 3, we present the accuracy of each mapping network, highlighting those concepts that are relevant to the corresponding main network. Relevant concepts were typically extracted with the highest accuracy among all concepts, and non-relevant concepts were extracted with the lowest. For example, the extraction of ∃has.ReinforcedCar (resp. RuralTrain) achieved better results for the main network $\mathsf{NN_A}$ (resp. $\mathsf{NN_C}$) for which it is relevant. Also worth noting is, for example, the fact that concept ∃has.OpenRoofCar, which is not relevant for any of the three main networks, was not extractable from neither. There are situations where some relevant concepts turn out to be redundant given other concepts that are extractable, i.e. that the main network has learned, rendering them *less extractable*. This is the case with ∃has.LongWagon in $\mathsf{NN_B}$, which is redundant given LongTrain and PassengerTrain. Non-relevant concepts might also be learned, due to a bias in the dataset. This is the case with LongTrain in $\mathsf{NN_C}$, since most images of TypeC trains are long trains.

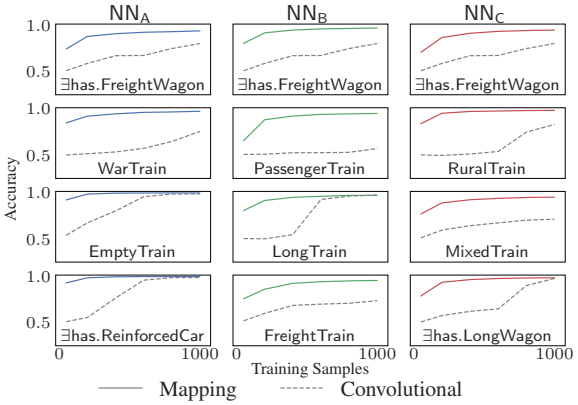

Figure 4: Accuracy of mapping networks and convolutional neural networks when trained to identify the same concepts.

4.2 Cost of the Mappings

The results from the previous experiment suggest that it is possible to extract human-defined concepts from neural networks models, as long as those concepts are relevant to the task performed by that neural network. However, apart from our interest in better understanding the information encoded in the architecture of a neural network, it is necessary to assess whether training a neural network to predict a given concept based on the activations of another neural network is practical, taking into account the amount of required training data, and the size and accuracy of the resulting model.

To assess the cost of training the mapping networks, we contrasted the results obtained by extracting the 4 relevant concepts with higher accuracy from each main network, out of the 11 previously considered, with those obtained by training convolutional neural networks using the images in the dataset to predict those same concepts. For our main hypothesis to be practical, training the mapping networks should require less training data than training the convolutional neural networks. The accuracy of the networks developed in both settings is compared when different amounts of training data are available, using 20% of the available data for validation, and a test set of 1 000 samples. The validation set was used to choose between the two different models of mapping networks being considered, and to choose between 22 different models of convolutional neural networks, each similar to or simpler than those used in the main networks.

In Figure 4, we plot the accuracy of the neural networks developed in both settings, against the amount of available training data for each considered concept. Mapping networks typically achieve higher accuracy values than the corresponding convolutional neural networks, especially when the number of available training examples is low. The results suggest that the use of the activations of a neural network to predict relevant concepts yields smaller models achieving higher accuracies, and requiring less training data.

This experiment shows two main benefits that stem from using the activations of a trained neural network to predict relevant concepts. First, the overhead caused by the development and use of a mapping network is minimal, since the model is much simpler than what otherwise would have been necessary. Then, the amount of required labeled data for training with the same accuracy value is usually smaller when extracting a concept using a mapping network.

This seems to further suggest that the information encoded in the activations of a neural network can, in a sense, be closer to the human-defined concepts than the information present in the input features, assuming that the concepts are relevant to the task of this neural network.

4.3 Meaning of the Extracted Concepts

Having evidence that it is possible to extract human-defined concepts from the activations of a trained neural network, and that it might only require few labeled data to do so, we address the natural concern related to the use of mapping networks: "Do the extracted concepts resemble our understanding of those concepts? Or are the neural networks finding meaningless correlations in the activations of the many different neurons which are being fed as input?".

To answer these questions, we employed the occlusion procedure described in (Zeiler and Fergus 2014), which works by systematically occluding different portions of an input image with a grey square and observing how the output of a neural network changes relative to the position of the grey square, allowing for a visualization of the probability of the correct class as a function of the position of the occluding grey square. We apply this procedure by feeding the activations of the occluded images to the mapping networks, allowing for an estimation of how mapping networks' react to the main networks' input features.

Figure 5 shows the images of Figure 1 and the resulting attribution map obtained through this procedure. In the first example, we tested a mapping network on $\mathsf{NN_A}$ identifying whether a train has a *passenger car*, represented as a wagon containing a circle. The output of the mapping network drops when the first wagon – the *passenger car* – is occluded. In the second example, we used $\mathsf{NN_B}$ to assess a mapping network trained to identify *freight trains*. The image depicts a *freight train*, given that it has two wagons containing commercial goods – represented by a hexagon and a diamond. The mapping network properly identifies this concept, changing its output only when one of the two commercial wagons is covered. The third example shows an image of a train with two long wagons. Using a mapping network on $\mathsf{NN_C}$ trained to identify long wagons present in its input,

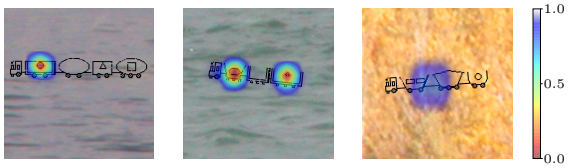

Figure 5: Map of correct class probability as a function of the position of a occluding square for three mapping networks.

only when the grey square occludes partially both wagons does the output of the mapping network decreases.

We tested the 12 mapping networks chosen in the previous Section, trained using 800 samples, by applying the above-described procedure to generate 1 200 attribution maps of 100 different images. Through visual evaluation, we were able to precisely understand the resulting attribution map given the concept being identified by a mapping network in 94% of the images. This experiment provides strong evidence that mapping networks are correctly localizing the concepts that they are trained to identify and appropriately reacting to the visual features that embody each concept.

4.4 Origin of the Extracted Concepts

So far, we have been using as input to the mapping networks all activations fed to and produced in the dense part of the main network, which might not always be feasible, e.g., due to memory constraints. Pinpointing which neurons' activations are required to extract a given concept is a crucial task, since it allows for the development of mapping networks that are smaller in input size, potentially achieving higher accuracies, besides providing useful information regarding where on a neural network each concept is more prominent.

In most non-trivial neural networks, due to their size, it is unfeasible to explore all possible sets of input features to train a mapping network. We designed a procedure – *Input Reduce* – that searches for a set of features with a given maximum cardinality that allows for a mapping network to be trained with the greatest possible accuracy. The procedure, c.f. Figure 6, keeps track of a set of best performing features, i.e., the smallest set of features that results in a mapping network achieving the highest accuracy, while iterating through the layers of a neural network, from output to input. At each layer a set of input features, composed by all neurons of the current layer together with the known best performing set of features, is considered. This set is iteratively reduced through the use of a feature ranking method and tested by building and evaluating mapping networks that use it. A patience parameter is used to speed up this process, anticipating the move to the next layer. The procedure returns the best performing set of features found, either after the last layer of the neural network is processed, or when no new best performing set of features is found in a given layer.

We tested the Input Reduce procedure for the 4 relevant concepts previously selected for each of $\mathsf{NN_A}$, $\mathsf{NN_B}$, and $\mathsf{NN_C}$, using a training set of 800 samples, a validation set of 200 samples, and a test set of 1 000 samples. The test was performed using the Input Perturbation Feature Importance algorithm (Heaton et al. 2017), a patience value of 8,

Input : $main_{nn}$ – main network architecture
map_{nn} – mapping network architecture
$data_{tr}$ – training data of mapping networks
$data_{val}$ – validation data of mapping networks
$ranker$ – feature ranking algorithm
$patience$ – amount of steps to wait before changing layer if no progress on validation accuracy
$remove_{\%}$ – % of features to remove each step
max_{feats} – maximum number of features
Output: a set of inputs

begin
  $feats_{best} \leftarrow \emptyset$
  $acc_{best} \leftarrow 0$
  for $layer$ **in** $reversed(main_{nn}.layers)$ **do**
    $curr\_feats \leftarrow feats_{best} \cup layer.neurons$
    $curr\_patience \leftarrow patience$
    $new\_solution \leftarrow$ false
    while not $curr\_feats.is\_empty()$ **do**
      $map_{nn}.train(data_{tr}[curr\_feats])$
      $acc \leftarrow map_{nn}.eval(data_{val}[curr\_feats])$
      if *(*$acc > acc_{best}$ **or** *(*$acc == acc_{best}$ **and**
       $|feats_{best}| > |curr\_feats|$*))* **and**
       $max_{feats} \geq |curr\_feats|$ **then**
        $acc_{best} \leftarrow acc$
        $feats_{best} \leftarrow curr\_feats$
        $curr\_patience \leftarrow patience$
        $new\_solution \leftarrow$ true
      else
        $curr\_patience \leftarrow curr\_patience - 1$
        if $patience < 0$ **then break**
      $rank \leftarrow$
       $ranker(map_{nn}, data_{tr}, data_{val}, curr\_feats)$
      $curr\_feats.remove(rank, remove_{\%})$
    if not $new\_solution$ **then**
      return $feats_{best}$
  return $feats_{best}$

Figure 6: Input Reduce procedure for feature selection.

	Output Concept	Dense Layers		Input Reduce	
		Accuracy	Features	Accuracy	Features
NN_A	∃has.FreightWagon	0.9367	10480	0.9263	453
	WarTrain	0.9719	10480	0.9930	4
	EmptyTrain	0.9937	10480	0.9942	2
	∃has.ReinforcedCar	0.9950	10480	0.9928	4
NN_B	∃has.FreightWagon	0.9676	10464	0.9629	2374
	PassengerTrain	0.9485	10464	0.9433	1107
	LongTrain	0.9670	10464	0.9701	534
	FreightTrain	0.9523	10464	0.9493	1247
NN_C	∃has.FreightWagon	0.9459	10608	0.9500	519
	RuralTrain	0.9820	10608	0.9916	7
	MixedTrain	0.9484	10608	0.9750	14
	∃has.LongWagon	0.9813	10608	0.9814	12

Table 1: Comparison of the accuracy of mapping networks developed using all activations fed to and produced in the dense part of their main network, and using the features resulting from the Input Reduce procedure.

$\mathsf{LongTrain}(\mathsf{i}_1)$
$\mathsf{FreightTrain}(\mathsf{i}_1)$
$\mathsf{LongFreightTrain} \equiv \mathsf{LongTrain} \sqcap \mathsf{FreightTrain}$
$\mathsf{TypeB} \equiv \mathsf{PassengerTrain} \sqcup \mathsf{LongFreightTrain}$

Figure 7: Input image and accompanying justification.

and removing 20% of the features at each step, with a maximum feature set size equal to the number of all activations fed to and produced in the dense part of their main network. Although this procedure does not provide formal guarantees regarding the resulting set of features, our empirical results, shown in Table 1, show that all of the mapping networks decreased their input size. On average, the mapping networks trained with the set of selected features required only 5% of the features achieving similar accuracy values. This is observable in the second example in Table 1, where the mapping network trained to identify *war trains* increased its accuracy and substantially reduced its input dimensionality when trained using the features obtained by Input Reduce.

5 The Justifications

The main motivation for extracting human-defined concepts from the activations of ensembles of neurons of a neural network was to be able to leverage on the ontology used to define those concepts to produce justifications for the neural network's output in a symbolic and declarative way. In a nutshell, if we add the observations produced by the mapping networks to the ABox of the ontology, a justification would be a minimal subset of the ontology that entails a fact representing the output of the network.

More formally, given an ontology $\mathcal{O} = < \mathcal{T}, \mathcal{A} >$, an ABox $\mathcal{A}'$ composed of facts describing the extracted concepts, and a formula (typically an atomic fact) φ representing the output of the neural network, a justification $\mathcal{J}$ for φ given $\mathcal{O}$ and $\mathcal{A}'$ is a subset of $\mathcal{O}' = < \mathcal{T}, \mathcal{A} \cup \mathcal{A}' >$ such that $\mathcal{J} \models \varphi$ and for all $\mathcal{J}' \subset \mathcal{J}$, $\mathcal{J}' \not\models \varphi$. The last requirement ensures minimality so that justifications only contain axioms that are necessary to support the entailment. Note that there may be more than one minimal justification.

Consider the scenario where we are analyzing $\mathsf{NN_B}$, the main neural network trained to identify trains of *type B*, for which we developed mapping networks for the concepts PassengerTrain, LongTrain, and FreightTrain. When we feed image (i_1) depicted in Figure 7 to this neural network, it correctly classifies it as being of *type B*, i.e., $\varphi = \mathsf{TypeB}(\mathsf{i}_1)$. By only taking into account the ontology, i.e., before looking at the extracted concepts, all we could infer is that the train in the picture is either a passenger train or a long freight train, but we would not be able to tell which. However, once we observe that the mapping networks were able to extract the concepts FreightTrain and LongTrain, though not PassengerTrain, then we can build the ABox:

$$\{\neg\mathsf{PassengerTrain}(\mathsf{i}_1), \mathsf{LongTrain}(\mathsf{i}_1), \mathsf{FreightTrain}(\mathsf{i}_1)\}$$

and find a justification for $\varphi = \mathsf{TypeB}(\mathsf{i}_1)$, depicted in Figure 7. This justification shows that the input of the neural

	All Correct	Some Correct	None Correct	No Justifications
NN_A	85.5%	14.3%	0.2%	0.0%
NN_B	94.2%	2.1%	0.7%	3.0%
NN_C	90.6%	8.9%	0.1%	0.4%

Table 2: Summary of justifications for each main network.

network was classified as a train of *type B* because it was identified to be both a *long train* and a *freight train*; and trains being both *long* and *freight* are *long freight trains*, and *long freight trains* are of *type B*. Naturally, these justifications could be translated to natural language before being presented to the user, e.g. as in (Androutsopoulos, Lampouras, and Galanis 2013).

To evaluate the justifications produced through this method, we employed a Description Logics axiom pinpointing algorithm described in (Horridge 2011). We used the mapping networks developed in Section 4.4 and, for each, we sought justifications for the output of 1 000 images containing train representations of the type they are trained to identify. The result for each image could fall into one of four cases: –all justifications produced were correct i.e., only used concepts correctly extracted; –some justifications were correct, but some were incorrect i.e., used some concept incorrectly extracted; –none of the produced justifications were correct; –no justifications were produced. Incorrect justifications can either be due to absent concepts that were incorrectly extracted, or present concepts that were not extracted. Absent justifications can either be due to the incorrect concept extraction, but could also be due to a poor choice of relevant concepts to be extracted.

The results, summarized in Table 2, are quite positive as the method was able to find correct justifications in most cases. Whereas the experiments we conducted benefited from a controlled environment, even if the data and methods ensure that the results are valid, when employed in less controlled environments it is expectable that things may become more difficult. Either because of errors in the concept extraction, of poor choice of concepts to be extracted, we could end up not only with incorrect justifications, no justifications, or even obtain inconsistent justifications. There are nevertheless known methods involving abductive reasoning – e.g. minimally adding the required missing observations to be able to justify the output – belief revision – e.g. minimally removing observations to restore consistency – or even paraconsistent reasoning, that could be employed, which nevertheless fall outside the scope of the present paper.

6 Related Work

Popular approaches to interpretability of neural networks are saliency and attribution methods (Ancona et al. 2018; Zeiler and Fergus 2014), where information regarding the importance of each input feature to a given prediction is produced. Other methods try to build more concise representations of neural networks, e.g., (Zhang et al. 2018). Despite their relevance, they do not provide any type of language to describe their results. Proxy-based methods (Ribeiro, Singh, and Guestrin 2016; Zilke, Mencía, and Janssen 2016) typically substitute a model for one that is interpretable by design and that behaves similarly to the original model. In contrast, our justifications don't require changing or substituting the original model, which might not always be feasible.

In the field of neural-symbolic integration there has also been work towards increasing the interpretability of neural networks. For example, (Hitzler, Hölldobler, and Seda 2004) seeks interpretability of neural networks by design since they are built to compute propositional logic programs. However, they miss on the neural network's capacity to learn from examples. In (Sarker et al. 2017), the authors propose the use of ontologies as background knowledge to help extract formulae that explain neural networks, but their method is somehow limited as it only encodes the input-output behavior of the networks, and requires that data be labelled with the required concepts.

7 Conclusions

In this paper, we linked neural networks and ontologies with the objective of obtaining justifications for a neural network's output. To this end, we assessed the implications and benefits of the use of mapping networks – neural networks built to predict human-defined concepts from the activations of ensembles of neurons from a neural network. Through experimental evaluation, we showed that the developed mapping networks are small in size, requiring less computational power, training time, and training data than if we were to develop a neural network without leveraging on the activations of the neural network which output we want to justify.

We conclude that it is possible to leverage on the knowledge hidden in the architecture of a neural network and to use that knowledge to establish mappings to concepts from an ontology, in order to extract symbolic justifications for a neural network's output that are human-comprehensible. This allows us to peek into the concepts that a neural network has learned, increasing its interpretability.

To the best of our knowledge, our work is the first to leverage the representations encoded in the architecture of a neural network to obtain justifications for its output using human-understandable concepts defined through an ontology, without changing or retraining the neural network.

It is important to note that the justifications obtained through our method are not necessarily a representation of how neural networks *really* reach their outputs, but rather plausible and understandable justifications for how they might have achieved their results, based on human-understandable concepts. The process somehow resembles a similar human behavior: sometimes we make choices, and only when compelled to explain the reason behind them, do we stop for a moment and try to rationalize them. Independently of their nature, this seems to be a relevant step towards bridging neural and symbolic Artificial Intelligence, with an immediate impact in the search for Explainable AI.

In the future, we wish to explore in-depth the extraction of justifications based on the ontology and the results of the developed mapping networks, investigate the usage of this method to search for learned biases in trained neural networks, and as a way to search for errors, missing axioms, or missing relations in ontologies.

Acknowledgments

The authors would like to thank Ludwig Krippahl for the helpful discussions while working on this paper, and the anonymous reviewers for their insightful feedback. We would also like to acknowledge and thank the support provided by Calouste Gulbenkian Foundation through its *"New Talents in AI"* program, and by FCT through project ABSOLV (PTDC/CCI-COM/28986/2017), and strategic project NOVA LINCS (UIDB/04516/2020).

References

Alain, G.; and Bengio, Y. 2017. Understanding intermediate layers using linear classifier probes. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Ancona, M.; Ceolini, E.; Öztireli, C.; and Gross, M. 2018. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Androutsopoulos, I.; Lampouras, G.; and Galanis, D. 2013. Generating Natural Language Descriptions from OWL Ontologies: the NaturalOWL System. *J. Artif. Intell. Res.* 48: 671–715.

Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Debiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; Józefowicz, R.; Gray, S.; Olsson, C.; Pachocki, J.; Petrov, M.; de Oliveira Pinto, H. P.; Raiman, J.; Salimans, T.; Schlatter, J.; Schneider, J.; Sidor, S.; Sutskever, I.; Tang, J.; Wolski, F.; and Zhang, S. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. *CoRR* abs/1912.06680.

Berners-Lee, T.; Hendler, J.; and Lassila, O. 2001. The Semantic Web. *Scientific American* 284(5): 34–43.

Biran, O.; and McKeown, K. R. 2017. Human-Centric Justification of Machine Learning Predictions. In Sierra, C., ed., *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 1461–1467. ijcai.org.

Bojarski, M.; Testa, D. D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L. D.; Monfort, M.; Muller, U.; Zhang, J.; Zhang, X.; Zhao, J.; and Zieba, K. 2016. End to End Learning for Self-Driving Cars. *CoRR* abs/1604.07316.

de Sousa Ribeiro, M.; Krippahl, L.; and Leite, J. 2020. Explainable Abstract Trains Dataset. *CoRR* abs/2012.12115.

Gkatzia, D.; Lemon, O.; and Rieser, V. 2016. Natural Language Generation enhances human decision-making with uncertain information. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics.

Graves, A.; Mohamed, A.; and Hinton, G. E. 2013. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, 6645–6649. IEEE.

Hassabis, D.; Chu, C.; Rees, G.; Weiskopf, N.; Molyneux, P. D.; and Maguire, E. A. 2009. Decoding neuronal ensembles in the human hippocampus. *Current biology : CB* 19(7): 546–554.

Heaton, J.; McElwee, S.; Fraley, J. B.; and Cannady, J. 2017. Early stabilizing feature importance for TensorFlow deep neural networks. In *2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017*, 4618–4624. IEEE.

Hitzler, P.; Bianchi, F.; Ebrahimi, M.; and Sarker, M. K. 2020. Neural-symbolic integration and the Semantic Web. *Semantic Web* 11(1): 3–11.

Hitzler, P.; Hölldobler, S.; and Seda, A. K. 2004. Logic programs and connectionist networks. *J. Appl. Log.* 2(3): 245–272.

Horridge, M. 2011. *Justification based explanation in ontologies*. Ph.D. thesis, University of Manchester, UK.

Huang, J.; Li, J.; Yu, D.; Deng, L.; and Gong, Y. 2013. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, 7304–7308. IEEE.

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Li, F. 2014. Large-Scale Video Classification with Convolutional Neural Networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, 1725–1732. IEEE Computer Society.

Kim, B.; Wattenberg, M.; Gilmer, J.; Cai, C. J.; Wexler, J.; Viégas, F. B.; and Sayres, R. 2018. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 2673–2682. PMLR.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kratzert, F.; Herrnegger, M.; Klotz, D.; Hochreiter, S.; and Klambauer, G. 2019. NeuralHydrology - Interpreting LSTMs in Hydrology. In Samek, W.; Montavon, G.; Vedaldi, A.; Hansen, L. K.; and Müller, K., eds., *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700 of *Lecture Notes in Computer Science*, 347–362. Springer.

Lai, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Recurrent Convolutional Neural Networks for Text Classification. In Bonet, B.; and Koenig, S., eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, 2267–2273. AAAI Press.

Larson, J.; and Michalski, R. S. 1977. Inductive inference of VL decision rules. *SIGART Newsl.* 63: 38–44.

Lee, K.; Turner, N. L.; Macrina, T.; Wu, J.; Lu, R.; and Seung, H. S. 2019. Convolutional nets for reconstructing neural circuits from brain images acquired by serial section electron microscopy. *CoRR* abs/1904.12966.

Nair, V.; and Hinton, G. E. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In Fürnkranz, J.; and Joachims, T., eds., *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, 807–814. Omnipress.

Olmos, A.; and Kingdom, F. A. A. 2004. A Biologically Inspired Algorithm for the Recovery of Shading and Reflectance Images. *Perception* 33(12): 1463–1473.

Pieters, W. 2011. Explanation and trust: what to tell the user in security and AI? *Ethics and Information Technology* 13(1): 53–64.

Quiroga, R. Q.; Reddy, L.; Kreiman, G.; Koch, C.; and Fried, I. 2005. Invariant visual representation by single neurons in the human brain. *Nature* 435(7045): 1102–1107.

Radovic, M.; Adarkwa, O.; and Wang, Q. 2017. Object Recognition in Aerial Images Using Convolutional Neural Networks. *J. Imaging* 3(2): 21.

Redmon, J.; Divvala, S. K.; Girshick, R. B.; and Farhadi, A. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 779–788. IEEE Computer Society.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In Krishnapuram, B.; Shah, M.; Smola, A. J.; Aggarwal, C. C.; Shen, D.; and Rastogi, R., eds., *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 1135–1144. ACM.

Sarker, M. K.; Xie, N.; Doran, D.; Raymer, M.; and Hitzler, P. 2017. Explaining Trained Neural Networks with Semantic Web Technologies: First Steps. In Besold, T. R.; d'Avila Garcez, A. S.; and Noble, I., eds., *Proceedings of the Twelfth International Workshop on Neural-Symbolic Learning and Reasoning, NeSy 2017, London, UK, July 17-18, 2017*, volume 2003 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 815–823. IEEE Computer Society.

Segler, M. H. S.; Kogej, T.; Tyrchan, C.; and Waller, M. P. 2018. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Central Science* 4(1): 120–131.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; Chen, Y.; Lillicrap, T. P.; Hui, F.; Sifre, L.; van den Driessche, G.; Graepel, T.; and Hassabis, D. 2017. Mastering the game of Go without human knowledge. *Nat.* 550(7676): 354–359.

Sultana, F.; Sufian, A.; and Dutta, P. 2019. Advancements in Image Classification using Convolutional Neural Network. *CoRR* abs/1905.03288.

Topol, E. J. 2019. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine* 25(1): 44–56.

Ye, L. R.; and Johnson, P. E. 1995. The Impact of Explanation Facilities on User Acceptance of Expert Systems Advice. *MIS Quarterly* 19(2): 157–172.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and Understanding Convolutional Networks. In Fleet, D. J.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., eds., *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, volume 8689 of *Lecture Notes in Computer Science*, 818–833. Springer.

Zhang, Q.; Cao, R.; Shi, F.; Wu, Y. N.; and Zhu, S. 2018. Interpreting CNN Knowledge via an Explanatory Graph. In McIlraith, S. A.; and Weinberger, K. Q., eds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 4454–4463. AAAI Press.

Zilke, J. R.; Mencía, E. L.; and Janssen, F. 2016. DeepRED - Rule Extraction from Deep Neural Networks. In Calders, T.; Ceci, M.; and Malerba, D., eds., *Discovery Science - 19th International Conference, DS 2016, Bari, Italy, October 19-21, 2016, Proceedings*, volume 9956 of *Lecture Notes in Computer Science*, 457–473.