Reasoning About What Has Been Learned

Neural-Symbolic Integration for Explainable Artificial Intelligence

Benedikt Wagner

City, University of London

Department of Computer Science

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

January 2022

Dedicated to the memory of Dr. Martin Wagner, whose love, compassion, and perseverance always served as a guide.

Acknowledgements

First of all, I would like to thank my supervisor, Prof. Artur Garcez. Throughout this process he has provided me with useful feedback, provocative ideas, and overall support. His enthusiasm and dedication to the field of Neural-Symbolic computing have played a significant role in this work.

Furthermore, I would like to thank my family, without whom this would not have been possible. I am especially grateful to Barbara, Konstantin, and Lisa, who have been a source of stability and support in my life which enabled me to pursue this work.

Also, I would like to express my gratitude to Adam and Sofoklis for their discussions and feedback regarding all the various ideas that sprang to mind at the time.

Lastly, I would like to thank Dr. Tarek Besold who sparked my interest in this work and has devoted considerable time and effort to providing me with a range of opportunities that culminated in this work.

Abstract

We investigate the potential of Neural-Symbolic integration to reason about what a neural network has learned. By undertaking a systematic study of the literature on explainable Artificial Intelligence, we propose a new taxonomy that showcases the methods in a structured manner and enables the integration of earlier work. Initially, two promising symbolic-inspired explainability methods for deep neural networks are evaluated. We examine the limitations of a concept-based XAI approach and expand the applicability of the method to a new data domain. The approach is extended by integrating ontology querying to provide more comprehensive explanations. Having examined the internal representation of the network, we investigate the appropriateness of a decision tree extraction method to explain the inner operations. Specifically, we apply the original proposal to increasingly complex visual tasks while extending the method to provide a deeper understanding of the extracted trees. In order to overcome the limitations identified in the examined methods, we propose a novel Neural-Symbolic approach to explainability by exploring the connection between conceptual representation and expressive first-order logic operators for intuitive and powerful explanations corresponding to human reasoning. The relevant framework is adapted from Logic Tensor Networks and modified to add a continual interactive explanation mechanism for model-agnostic querying and constraining. We examine the reasoning capabilities of neural networks that have been trained using the framework to identify the necessary requirements for a model to be integrated effectively. This allows us to establish what constitutes valuable background information for improving deduction as compared to previously published benchmarks. Using the novel interactive framework, we present a method for acting on information extracted by any XAI approach to prevent the model from learning unwanted behaviour and biases and show how the method can be applied to quantitative fairness. Compared with another constraint-based neural network approach, we demonstrate improved accuracy, while maintaining fairness based on two common fairness metrics. Furthermore, we incorporate the initial work on concept groundings into the new framework to facilitate comprehensive conceptual and logic-based explanations of the black box model (i.e., CNN and transformer). It demonstrates that model explanations can retain truthful representations of operations and internal representations, passing a benchmark proposed for *truly explainable AI.*

Contents

List of Figures

Chapter 1

Introduction

Recent years have seen a vast increase in the popularity of Artificial Intelligence-based systems, including Neural Networks and similar approaches. As a result, these approaches are now pervasive in many aspects of our daily lives, including the use of media, mobile phones, finance, as well as health care, as they are becoming more sophisticated in their capacity to solve a wider range of problems. Artificial intelligence approaches provide a variety of benefits, including process cost reduction, repeatability, improved efficiency, effective decision making, as well as a means for developing new products and services. In addition to disrupting many industries, artificial intelligence is also spawning new technologies and trends that can transform our lives, such as self-driving cars and intelligent assistants.

As a field of research, Artificial intelligence dates back more than half a century, but has since seen exponential growth in research output due to a wide range of factors, including increasing computational capacity, greater affordability, better generalizability of models, and the availability of unprecedented amounts of data (Rosenblatt, 1958; LeCun et al., 2015). The combination of these factors led to significant improvements in predictive performance across a variety of tasks. Complex Machine Learning (ML) models, such as Neural Networks (NN), exhibit powerful classification, prediction, and optimisation capabilities, but they lack transparency and comprehensibility. Such models have the advantage of being able to discriminate using an arbitrary function in an end-to-end manner, with the optimisation based on an input-output mapping, and the model having the flexibility to adjust a vast number of parameters as required. A weakness of the models' complex and distributed representation is that they can be considered black boxes in terms of interpretability since the parameter values do not directly provide any insight into the underlying logic and reasoning of the model. Historically, accuracy and interpretability have been suggested to be unavoidable tradeoffs. However, recent research has identified more effective methods for dealing with this trade-off.

Many situations demand more from an AI system than simply providing a suitable outcome. Rather, we may wish to understand more about the reasoning process used to arrive at a decision. The commonly used approach of evaluating accuracy on a test set may serve as an initial metric to measure the performance of an AI system; however, the system may still learn decision-making procedures that may be inherently inappropriate for the task (Rudin, 2019). This phenomenon may be due to the limited scope of information available in the dataset, which constrains the applicability of the learned reasoning steps.

With the advancement of artificial intelligence, economic growth is expected to accelerate, transforming society based on the way we interact with technology. Accordingly, regulators and stakeholders expect that AI will be inclusive and beneficial to everyone. The UK Parliament House of Lords AI select committee (House of Lords and Select Committee on Artificial Intelligence, 2018), for instance, mentions in its report that, "we believe it is not acceptable to deploy any artificial intelligence system which could have a substantial impact on an individual's life unless it can generate a full and satisfactory explanation for the decisions it will take".

As part of current initiatives undertaken by regulatory authorities at different levels (such as, e.g., the EU General Data Protection Regulation 2016/679 (European Union, 2016)) as well as societal discussions regarding a desire for greater transparency and accountability in automated decision-making, research into better interpretable and comprehensible methods and systems in Artificial Intelligence (AI) is expanding. Therefore, there has been a wide variety of methods introduced in recent years, leading to a diverse mixture of approaches geared towards understanding the behaviour of a model. A number of approaches aim to extract decision-related information from complex models into simple models, thus achieving considerable interpretability at the expense of confined performance that results from introduced model constraints, while others provide explanations by describing representative cases (e.g. Frosst and Hinton (2017); Kim et al. (2018)). A number of methods reduce the dimensionality of a model representation in order to facilitate visual analysis of its inherent features (e.g. Van Der Maaten et al. (2009)). In other approaches, a local explanation is provided, which restricts the explanation to specific instances that serve as proxies for the model's behaviour (e.g. Ribeiro et al. (2016)). Consequently, we have devoted a considerable portion of the following work to providing a holistic review and taxonomy of the wide range of explainbility approaches, a necessity in such an expanding and dynamic field of research.

The initial focus will be on extending existing explainability methods and exploring their limitations so as to reveal some fundamental weaknesses. By doing so, we emphasise the human's involvement in an explainable AI (XAI) system. Here, it is useful to draw inspiration from the way people communicate explanations in their daily lives. An objective for explanations is to be intuitive and thus similar to human explanations, which have a proven track record of effective communication. Nevertheless, they must be truthful to the representations and operations of the models, since only sound explanations should be accepted. Often, as will be demonstrated, the coherence of representation as well as operation is not sufficiently addressed in current XAI approaches.

Furthermore, we consider explainable AI to be more than a fixed description of a static system. Neural-symbolic approaches to XAI should be an interactive, continuous process that allows not only to describe the system but also to convey information back to the system, enabling the user to act on the system's description.

1.1 Illustrating why XAI is needed

The majority of current Machine Learning systems have a substantial reliance on statistical and data-driven strategies in common in order to perform a large variety of tasks. This pivotal role of large amounts of data as input, processed using sophisticated statistical techniques without explicit generation of interpretable knowledge along the way, results in incomprehensible representation and a lack of understanding of internal operations. The corresponding systems are opaque ”in the sense that if one is a recipient of the output of the algorithm (the classification decision), rarely does one have any concrete sense of how or why a particular classification has been arrived at from inputs. Additionally, the inputs themselves may be entirely unknown or known only partially” (Burrell, 2016). Due to the opacity of this process, understanding the outcomes of a system's decision-making processes presents a challenge. Different factors contribute to the need for explainability of models, particularly when impacting safety-critical areas. Caruana et al. (2015) describes a representative example of interpretability in the medical domain. In an innovative piece of medical research, they developed a neural network model to calculate the likelihood of severe illness leading to death for patients entering a hospital with pneumonia. The objective was to determine quickly whether a patient needed to be admitted to the hospital or sent home. It was found that the model predicted a patient's likelihood of death with surprising accuracy, but it was fortunately not implemented until further investigation was conducted. It became apparent from looking at the behaviour of the neural network model that it had inferred the practice that patients with pneumonia and asthma have a minimal risk of death. Research contradicted the conclusion drawn by the model, however it was the pattern derived from the training data that achieved high accuracy on the dataset. The reason behind this unusual pattern was the fact that patients with asthma were regularly directly admitted to intensive care, receiving immediate treatment and often succeeding in their treatment and therefore recovering well. As a consequence, the model could have caused significant harm if it had not been revised and the behaviour thoroughly investigated.

1.1.1 Goals of explaining ML systems

Aside from the regulatory necessity of integrating explainable methods into automated decision making, there are several benefits to gaining insight into the decision making process. The variety of possible scenarios and associated contexts is as diverse as the range of terminology associated with AI explainability and its multitude of applications and stakeholders. Nonetheless, prominent figures in the field of Machine Learning have voiced reservations regarding the necessity of explainable AI (LeCun et al., 2017), while others have questioned the validity of post-hoc explanations of black-box models in particular (Rudin, 2018).

The requisite characteristics of an explainable system are one primary cause of a lack of consensus; these properties derive directly from the varying application domains with differing necessity and specific purposes of explanation (including requirements for an explainable system). The purpose sets the tone for constraints and desiderata, and is therefore an important aspect when considering interpretability.

However, there are distinct applications where explanations may not be necessary. For instance, predictions which will not result in immediate decisions, or fields that are not safety-critical. In such instances, the consequences for an unacceptable result are not necessarily severe. In many instances, it is argued that the need for explanations is justified by the consequences of a model's output as well as the appropriate means to measure the model's performance in this context. Currently, most machine learning practices rely heavily on measuring the accuracy of the models based on a held-out test set. As part of the Machine Learning paradigm, we assume that the data collection process for training should correspond as closely as possible to the data encountered by the model in deployment. However, if this assumption does not

hold or should be violated in some way, there may be a benefit in being able to collect more information than merely performance metrics.
In some applications scenarios, for example, an explanation is necessary to justify a system and its behaviour, and it may be possible to find a satisfactory solution through a thorough analysis of the data distribution prior to training. Further, the specific type of use-case also determines the requirements regarding the coverage of the explanation. The following are typical explanation purposes motivating the use of explainable approaches (Adadi and Berrada, 2018):

- Explain to Justify: ensure that decisions are made correctly and increase trust in the system. The ability to query our model and discover implicit interactions so that we can gain an understanding of which features may be important in terms of the decisions made by the model. These explanations ensure the integrity of the model and prevent unwanted inclinations or differentiation.

- Explain to Control: improve the prevention of errors and the detection of flaws. Providing the capability to evaluate and validate any data point and how a model arrived at a particular decision in relation to individual entities. Key stakeholders should be able to demonstrate and comprehend that the model works as expected. These explanations help to ensure model transparency and prevent unwanted discrimination.

- Explain to Improve: in understanding the inner workings of a system, we can find ways to improve it. A model's interpretability can be beneficial to machine learning engineers, as it allows for improved maintenance and continuous learning of techniques across different tasks, based on an improved understanding of the model's behaviour.

- Explain to Discover: we want to learn new facts and gather information to gain knowledge and insights. Transparency will make such information accessible and allow for extrapolation of the discovered facts in the future.

Current XAI methods are typically tailored to a particular use case. We suggest that a truly explainable system that serves as a communication bridge will be capable of addressing all of the above use-cases collectively. Additionally, many approaches maintain a disconnect between the actual description and the goal of the explanation.

While the purpose of these methods may be to control against unwanted discrimination, current XAI methods simply display disparities without addressing them directly. Such methods are considered unidirectional tools and provide only descriptive information with no capability to act upon it.

1.2 Interpretable and Comprehensible Models

Before outlining the variety of solutions addressed in the literature regarding the facilitation of understandable decision making, it is crucial to clarify how interpretability and explainability are defined. Accordingly, we will examine different explainable AI methods with the aim of dissecting various aspects of interpretability and distinguishing different criteria for a model's explanation.

Interpretability is nowadays often referred to as an indicator that a human is capable of understanding the machine-learning model to an undefined extent. In contrast to symbolic representations, we may not be able to fully reconstruct the reasoning of a connectionist system. In reading the UK Government's report on the current state of AI in the UK, (House of Lords and Select Committee on Artificial Intelligence, 2018), another central issue becomes apparent. The report mentions that terminology varied considerably among practitioners. The terms transparency, interpretability, intelligebility, and explainability are often used interchangeably.

In machine learning literature, interpretability is loosely defined as the ability to explain or to provide the meaning in understandable terms to a human (Miller, 2017). The implicit assumption is that information representation is expressed in understandable terms constituting a self-contained explanation. Consequently, the explanation can be seen as an "interface between humans and a decision maker that is at the same time both an accurate proxy of the decision maker and comprehensible to humans" (Guidotti et al., 2018). Hence, interpretability refers to being able to better understand the decision policies of a machine-learned response function and ultimately explain the relationship between input and output variables in a way that is human-interpretable.

In general, an explanation is a set of statements that are usually formulated to describe a particular collection of facts that address the underlying causes, context, and consequences of those facts (Drake, 2018). As well as establishing rules or laws, the description may also clarify these rules or laws in relation to the objects or phenomena it examines. The theory of communication specifically proposes communication to be predominately symbolic (Littlejohn, 1977). Here, the so-called symbolic interactionism suggests that humans interpret and assign meaning to events through a set of symbols that are interrelated. Many years of research on communication have been built upon these fundamentals as they may establish a broad framework for all forms communications (Littlejohn, 1977).

Consequently, we will concentrate on two core components of desirable explanations: symbols and operations as relations between symbols. This perspective ties directly into the notion of neural-symbolic explanation, in which we intend to integrate the powerful learning capabilities of neural networks with the intuitive and expressive logical description using symbols and operators to serve as bridges of communication. The establishment of this communication bridge necessitates that information be revised and inserted in tandem with its extraction. Therefore, we propose several experiments for measuring the effectiveness of translating information into the network as well as extracting information on what the network has learned.

1.3 Objectives and Contributions

In this thesis, we shall investigate the potential of neural-symbolic approaches to reason about what a deep neural network has learned. Our objective here is to examine the multiple advantages that can be achieved from this approach to explainable artificial intelligence. Particularly, we are interested in examining its ability to derive model-inherent logic through the use of multiple levels of abstract representation. Methods currently in use for XAI usually rely on statistically driven explanations based on input features that fail to take into account the powerful representations learned by the model. We propose that an effective XAI method, as opposed to existing XAI approaches, should act as a lingua franca between an AI system and a human counterpart, rather than simply presenting information in an understandable manner. Since a neural-symbolic approach is presented as a communication bridge, we examine the effectiveness of the model as it relates to its ability to revise model behavior through constraints. As a result, a continuous human-machine interactive process for more understandable and desirable AI systems is enabled.

An analysis of the extensive literature on XAI methods with a variety of methodologies is followed by an investigation of two existing symbolic-inspired explainability methods for deep networks with an examination of their limitations. These will highlight the existing shortcomings of present XAI approaches. Afterwards, we argue for tighter neural-symbolic integration to help overcome the observed limitations and introduce our method as an interactive communication bridge. We further propose a novel approach to explainability using real-valued (fuzzy) logic training and evaluation of deep neural networks.

The contributions can be summarised as follows:

- We introduce a new taxonomy for explainable artificial intelligence that presents the methods in a systematic accessible manner and allows for the integration of earlier work.

- We investigate the limitations of the symbol grounding related TCAV method and extend the approach to new data domains. In order to arrive at more comprehensive explanations, the approach is further integrated with ontology querying.

- We study the limitations of the Soft Decision tree rule extraction method for explaining the operations within a neural networks. In addition, we present an adaptation of the approach to increasingly complex visual tasks. By restricting the model in specific ways, we allow for better understanding of the model.

- We propose a bottom-up Neural-Symbolic approach to explainability by integrating knowledge and data into a neural network and enabling post-hoc information extraction as well as targeted retraining based on knowledge revision.

- We implement and adapt a framework for neural networks to enable continual learning and iterative querying by caching learned representations and by using network querying in first-order logic to check for knowledge learned by the deep neural network.

- We examine the reasoning capabilities of the proposed framework and identify what constitutes valuable background information that contributes to improved deduction.

- We present a method for acting on information extracted by any XAI approach in order to prevent the model learning unwanted behaviour or bias. By leveraging an existing XAI method, namely SHAP (Lundberg and Lee, 2017), we demonstrate how our method can be used to identify and address undesired model behaviour. Furthermore, we demonstrate how bias can be identified using the querying mechanism of the proposed framework.

- We apply the proposed method and tool to the field of quantitative fairness in finance. We demonstrate improved accuracy in comparison with another state-of-the-art neural network-based approach (Padala and Gujar, 2020) across three datasets while retaining fairness based on two common fairness metrics.

- We demonstrate that the framework is capable of dissecting model explanations into complex concepts and facilitating interactive learning. It enables domain experts to learn about the data-inferred decision making process of large ML models by querying the model and defining constraints for further learning as part of an iterative process. In an iterative process, it is possible for domain experts to gain insight into the data-driven decision making processes of large ML models using abstract concepts integrated into expressive first-order logic queries and subsequently to set constraints to guide further learning.

1.3.1 Contributions

The following publications contain material in this thesis:

- Benedikt Wagner, Artur D'Avila Garcez, *Neural-symbolic Integration for Fairness in AI, AAAI 2021 Spring Symposium*, AAAI-MAKE 2021.

- Benedikt Wagner, Artur D'Avila Garcez, *Neural-Symbolic Integration for Interactive Learning and Conceptual Grounding, NeurIPS 2021*, Workshop on Human and Machine Decisions (WHMD 2021).

- Benedikt Wagner, Artur D'Avila Garcez, *Neural-Symbolic Reasoning under Open-World and Closed-World Assumptions, AAAI 2022 Spring Symposium*, AAAI-MAKE 2022.

- Roberto Confalonieri, Ludovik Coba, Benedikt Wagner, Tarek R. Besold, *A historical perspective of explainable Artificial Intelligence, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Volume 11 Issue 1, 2021.

The publication taking a historical perspective on *explainable AI* is related to the arguments presented in chapter 5 and the section 3.3. The publication on *Fairness* will be discussed in section 6.4. The publication on *Reasoning* is covered in section 6.3 and the work on *Conceptual Grounding* in section 6.5.

1.4 Structure

Considering the increasing awareness of the need for AI to be explainable, the question of how to accomplish such a goal has gained renewed attention. Our goal is to take advantage of the neural-symbolic approach to tackle this challenge. As we will outline in the following section, we propose to investigate this area because the neural and symbolic approaches are complementary, and this angle has not been widely explored in the context of explainability in AI.

Initially, we will provide a high-level introduction to particular terminology, concepts, and prerequisite knowledge in chapter 2. Subsequently, we cover a wide range of explainability approaches in order to provide a broader context. As a result of the lack of structure in the field of explainable AI, we create a taxonomy to introduce the wide range of methods in a systematic manner in chapter 3. Furthermore, this allows practitioners to categorise the vast range of approaches in an accessible manner. The structure of this taxonomy is derived from earlier studies on rule extraction (Andrews et al., 1995) in order to facilitate comparisons with existing approaches.

Following this, we examine two existing methods more closely and extend them in different ways in chapter 4. To begin with, we examine the TCAV method in detail, we translate the approach into a new domain of data in order to understand its potential limitations. More importantly, we integrate ontologies so that background knowledge can be included for more thorough explanations. The focus here is on the grounding of concepts that are similar to symbol grounding. In order to dissect the model explanation into various concepts, we use intermediate representations of the model to arrive at the groundings. However, while the approach provides a pathway toward gaining insights into the model, it lacks the operations and relations that are critical to providing expressive explanations.

Since we can only decompose model explanation into the sensitivities of classes with respect to concepts, we subsequently examine the feasibility of an alternative tree-based method with an approach that emphasises the explanation of internal operations. Using this approach, the model architecture will be distilled into a simpler format that will provide insight into how a model is constructed. In this case, the emphasis is placed on the operational representation by converting the explanation into a sequential structure similar to that of a neural network. Using this approach, we investigate limitations of the approach with respect to increasingly complex visual tasks and extend the method to the medical domain and introduce more intuitive explanation representations.

In order to overcome the limitations identified, we propose a more comprehensive neural-symbolic approach to explainability in chapter 5. This chapter emphasises the connection between conceptual foundations and expressive first-order logic operators for intuitive and powerful explanations corresponding to human reasoning. The relevant framework is extracted from Logic Tensor Networks and extended to add a continual explanation mechanism for model querying and constraining in chapter 6.

Our initial approach will be to explore the reasoning capabilities of a neural network that has been trained using differentiable fuzzy logic as a constraint. Hence, we will be able to identify the necessary requirements for the model to be integrated accordingly as well as identify potential explanation limitations. Following this, we will illustrate our novel approach by showing how it can accomplish fair classification. We compare our approach to other ML techniques that are capable of producing fair classification based on particular quantitative fairness notions. In this context, fairness refers to preventing the model of unjust treatment of people belonging to a protected group. The fact that our framework will remain model agnostic is one of its key contributions. In other words, this allows for any ML model to be integrated and trained in accordance with the proposed fairness constraints.

Lastly, we incorporate the initial work on grounding concepts into a new framework that allows for comprehensive conceptual and logical explanations of the black box model in chapter 7. It will be demonstrated that the explanations can retain truthful representation of operations and representations while achieving desireable intuitiveness.

Chapter 2

Background

In the following section, we will provide an overview of the fundamental background information as well as explanations of terminology used in subsequent chapters. We will keep the introduction of the background material to a level that shall make the thesis self-contained and provide a basis on which everything that follows shall be based. Nonetheless, we will refer to other works that provide a more comprehensive explanation of specific topics.

We will introduce the related work in two ways. The taxonomy introduced in chapter 3 will provide an overview of many common explanationability methods. As a result of the interdisciplinary nature of the experiments, their corresponding application domain, and variety of assessed approaches, it is didactically more appropriate to introduce related work within each section.

2.1 Symbolism

It has historically been possible to broadly classify Artificial Intelligence research into two main approaches, namely symbolism and connectionism. The computational theory of mind provides the basis for symbolic artificial intelligence, where the goal is to construct computational models that closely resemble cognitive processes (Piccinini and Bahar, 2013). These processes operate at an abstract conceptual level, as opposed to low-level sensory data, and therefore the syntactic relationship of a given symbol to others is of crucial importance over its semantic meaning. The distinction between a symbolic system and something other than a symbolic system is not as simple as it initially appears, since there is an element of ambiguity involved. In this thesis we will focus on logical systems in particular and investigate how they could potentially benefit for the sake of explainability.

In a logical system, there is a formal language, which can be interpreted through some form of semantic interpretation, as well as a set of deductive rules. Therefore, an interpretation refers to an assignment of meaning to the symbols of a formal language. Models in logic are used to interpret abstract entities in a way that leads to conclusions about whether a sentence relating such entities qualifies as true based on interpretation of that model. Using deductive rules, it is possible to deduce sentences that follow logically from another set of sentences.

A number of logical sentences form the so-called Knowledge-Base $\mathcal{K}$. There is a syntactic consequence that can be drawn from it $\mathcal{K} \vdash l$ when there is a rule that deduces l from $\mathcal{K}$. We further say that $\mathcal{K} \models l$ or $\mathcal{K}$ entails l if l is true for every interpretation in which all sentences in $\mathcal{K}$ are true.

We describe a deductive system as sound when $\mathcal{K} \vdash l$ implies $\mathcal{K} \models l$ and as complete when $\mathcal{K} \models l$ implies $\mathcal{K} \vdash l$. Hence, when a deductive system accurately describes the semantic relationship between statements and knowledge base, it is termed "sound and complete" (Odense, 2019).

We are now going to examine propositional logic, as it lays the fundamental principles of most logical systems. In this context, we employ a countable set of propositional variables $X_1, X_2, ...$, also referred to as atoms, as well as a set of connectives $\{\neg, \rightarrow, \wedge, \vee\}$. Well-formed formulas refer to logical statements that follow the rules of the formal system. Formulas or sentences that are well-formed here are either atoms or can be derived from other well-formed formulas. Additionally, we assign a truth value to each atom in order to determine whether the formula is true or false.

The operators serve as the pillar for logical sentences, where $\neg$ is the negation or *not*, $\wedge$ is known as conjunction or *and*, $\vee$ as disjunction or *or*, and $\rightarrow$ as implication. Throughout this thesis, we refer to the implications from left to right, as follows: *antecedent* on the left to the *consequent* on the right.

Logical systems provide inference rules that enable truth assignments to be deduced from the truth assignments of other sentences. *Modus Ponens* permits us to deduce information from the knowledge base $\{\phi \rightarrow \psi, \phi\}$, enabling us to derive ψ. *Conjunction* allows derivation of $\{\phi \wedge \psi\}$ from $\{\phi, \psi\}$. Using *Simplification*, it is possible to deduce ψ and ϕ from $\{\phi \wedge \psi\}$.

First-Order-Logic extends propositional logic through the use of logical and non-logical symbols. In addition to the connectives presented above, the logical symbols will also include the universal and existential quantifiers ($\forall$ and $\exists$). Moreover, we employ variables, constant symbols, functions and relations. Functions and relations also have an associated arity, which specifies the number of arguments they require. If each required argument is assigned to a variable, constant, or function, these are also called terms. The connection of terms with relation symbols produces what is known as an atomic formula, which builds the most fundamental form of well-formed formulas in first-order logic. We use quantifiers to construct well-formed formulas, which means if ψ is a well-formed formula, $\forall x \psi$ and $\exists x \psi$ are also. In this context, ψ is also referred to as *scope* of the quantifier. Quantifiers are labelled as *bound* if the variable is contained within the *scope*, otherwise they are labelled as *free*.

Any sentence that contains a universal quantifier is true if the *scope* holds true when the quantified variable is replaced with any element of the set of objects. A sentence that includes the existential quantifier is true if the *scope* is true for any element in the set of objects used to replace the quantified variable. Atomic formulas are considered grounded when the argument of the relation is a constant or a function of constants, i.e. when the arguments do not contain variables.

For an in-depth presentation of propositional and first-order logic, we refer the reader to Russell and Norvig (2010) (pp. 234-357).

2.1.1 Decision Trees

Originally, decision trees were developed to represent a deterministic symbolic system in an interpretable manner. Even though the information may be equivalent to a set of rules, the representation of information is easily comprehensible. Consequently, decision trees are frequently employed as part of explainability approaches that attempt to provide insight into how a model operates.

Decision trees are graphical representations of propositional statements, each represented by a node. Through internal nodes or split nodes, input data will branch out and end up in a terminal node or leaf node which provides the decision in classification scenarios.

Decision trees have the advantage of being nonlinear in nature, where simpler models such as linear regression and logistic regression may still fail to perform well. Modelling real-world phenomena typically requires features to interact with each other as well as nonlinear outcomes. In these scenarios, decision trees can serve as an effective but also comprehensible model.

Based on a training set of data, a variety of algorithms can be applied to grow a tree. Two widely used methods of building decision trees are based on information gain (Quinlan, 1992), and the classification and regression trees algorithm (CART) (Friedman, 2001).

In the event of a split, information gain is determined by the difference in average entropy before and after the split. Here, the entropy is a measure of uncertainty of P defined as: $H(P) = -\sum_{x \in X} P(x)\log(P(x))$, where P is the probability distribution over a finite set X. If input examples are associated with training labels, we can derive probability distributions that correspond to entropies based on the frequency of labels.

As a result of the division of a set of examples into two subsets based on a split, probability distributions corresponding to each subset will have different entropies. Afterward, we compute the size-weighted average of both entropies. It is calculated as $\frac{|X_1|}{|X|} H\left(X_1\right) + \frac{|X_2|}{|X|} H\left(X_2\right)$, for the subsets X_1 and X_2 from X. Subsequently, the information gain can be calculated by comparing this average with the prior entropy. With this method, we can iteratively build the tree and choose splits that maximise information gain at each node of the tree. As a result, this process can either be repeated until a predefined threshold of splits is achieved, or until only one class of label is present at a node, which is then added as a leaf node.

A tree-like structure resulting from the combined statements provides a user with a decision model that allows them to comprehend conditional statements at each step, and how the accumulation of such steps leads to a conclusion. Nevertheless, they are typically less accurate than vector-based models such as neural networks.

2.2 Connectionism

Rather than rules, well-formed sentences, or symbols, Artificial Neural Networks (ANN) are the essential component of the connectionist framework. Artificial neural networks are based on biological inspiration by using simple computational units known as neurons are connected in conjunction with each other by weights. Further, this type of modelling allows us to use low-level sensory information as inputs. We are especially interested in Deep Neural Networks (DNN) as their powerful applicability across a wide range of applications combined with the black-box nature of the model calls for greater transparency. A network is considered deep when it contains multiple layers and is therefore complex.

2.2.1 Artificial Neural Networks

Originally inspired by the nervous system (McCulloch and Pitts, 1943), the principle concepts between artificial and biological neurons are similar. Inputs are received by each neuron from its connected neurons. If that input passes a threshold, the neuron will fire, sending the signals to all of the other connected neurons (McCarthy, 1963). In essence, a neural network may be viewed as a parallel model composed of simple computational units. These units, referred to as neurons, are linked by weights which facilitate the transmission of information.

Rosenblatt (1958) introduced the perceptron, which is the most basic form of ANN. It consists of a single layer with one neuron predicting class based on a row of input data. The neuron in the perceptron is binarised, which means it is considered either *on* or *off* based upon a simple step function that establishes the threshold.

To be more precise, in a neural network each neuron has an associated output $O\left(x_i\right)$ and an associated input $I\left(x_i\right)$. The input is determined by the weighted sum of the connected neuron outputs:

$$I\left(x_i\right) = \sum_j w_{j,i} O\left(x_j\right) + b_i$$

where $b_i \in \mathbb{R}$ represents an additional parameter known as bias, and $w_{j,i}$ the weights from neurons j to i. The activation function here is a step function that determines the output where $f(x) = 1$ and $O\left(x_i\right) = 1$ if $I\left(x_i\right) \geq 0$ following from $w * x + b \geq 0$. Otherwise $O\left(x_i\right) = 0$. A large number of activation functions exist, also known as transfer functions, which are used to map input from the neuron to an output $g : \mathbb{R} \to \mathbb{R}$. Commonly used functions include the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$ and the Rectified Linear Unit $\text{ReLu}(x) = \max(0, x)$.

These ANN models are particularly popular because they can be adapted by means of a learning algorithm. The neural network essentially adapts to a given task based on some data by adjusting the weights and biases in order to improve on a certain metric (Russell and Norvig, 2010).

A typical example of such a metric is the mean squared error $E = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$ where Y_i represents the label associated with the example i and $\hat{Y}_i$ the predicted label produced by the model. Given a set of examples, gradient descent may be used to optimize the parameters of a model with respect to the error function. This can be achieved using a technique called backpropagation, which propagates errors along a sequence of layers inside the model.

ANNs that do not have a sequence of layers, such as the perceptron, are ineffective in solving tasks that require non-linear decision boundaries. A simple perceptron, for example, cannot replicate the XOR logical gate. However, by adding so-called hidden layers between input and output layers that use a non-linear activation function, it has been demonstrated to be a powerful approach to solving complex problem sets. In fact, ANNs with two hidden layers with nonlinear activation function can yield universal function approximators (Hornik et al., 1989), meaning that they could approximate any continuous function. When this additional layer is applied, the ANN is referred to as a Multilayer-Perceptron (MLP).

2.2.2 Convolutional Neural Networks

Despite the fact that MLPs are universal approximators, architectural tweaks have been made to ANNs in order to achieve more effective learning in specialised areas. Computer vision is one such area that demands spatial invariance since the location of objects in an image should not influence the decision-making process during inference. A convolutional kernel, also known as a filter, is used to exploit the symmetry of the data by restricting the connections of a hidden neuron to a small segment of the input data. A filter kernel can be thought of as a small array of weights, typically a 3x3 matrix, although it may vary in size. For each segment of the image, the filter is applied by calculating the dot product between the input values, in this case the pixel values, and the filter. Upon storing the output in an array, the filter is shifted across the image in such a way that each segment of the image is covered. As a result, a series of dot products based on image inputs and filters is produced, which is also called a feature map or activation map. As each filter moves across multiple segments of the input image, its weights remain fixed, which is known as parameter sharing. The filters themselves are usually learned through backpropagation and gradient descent. It is common to have multiple filters resulting in differing depths of output as a result of the computation of numerous feature maps.

Additionally, CNNs employ the so-called pooling layers for dimensionality reduction, also known as downsampling, in order to reduce the number of parameters in the created feature map or input. A pooling operation can move over segments of an entire input, but it does not contain an array of weights. By aggregating the values within the segment, the filter/kernel populates the output array. There are two prominent types of aggregation: maximum pooling and average pooling. The max pooling method selects the pixel that has the greatest value to pass to the output array. Averaging pooling computes the average pixel value within each segment to pass on to the output array.

Pooling and convolutional layers are also called partially connected in contrast to their fully connected counterparts. Here, the values of the pixels are not directly connected to the output nodes. As the name suggests, in a fully connected layer each node in the output layer is directly connected to every node in the previous layer.

This procedure is usually used to carry out a classification task based on the extracted features.

Fukushima (1980) and LeCun Yann et al. (1998) provide some of the foundational work for CNN. The ability to successfully backpropagate to identify patterns would lead to a much broader spectrum of architectures based on such initial concepts. The improved exploitation of computational resources for more effective training on large datasets has been a major driver for architectural changes. An example of such would be the GoogleNet architecture Szegedy et al. (2015) which introduced the concept of inception layers. This technique utilizes multiple filters simultaneously on each layer, allowing the neural network to learn the best weights during training and automatically select the most useful feature maps.

2.2.3 Recurrent Neural Networks

In spite of their considerable success in a variety of areas, ANNs and MLPs consider all input simultaneously and information only flows in one direction. It is also for this reason that such networks are referred to as feed-forward networks. This is due to the fact that the network considers a single input in isolation and does not consider temporal factors. Although in ANNs it has been assumed that inputs and outputs occur independently of one another, this assumption does not hold in temporal domains as the output at one time step may depend on that at the previous time step. This led to the introduction of models that allow recurrence through the use of self-connections. Through such self-connections, a recurrent neural network may function like a memory in that it considers information from past inputs. This has been shown to be particularly useful to model time series. Here, the objective is to predict the next n steps of a time series given the previous m steps. As a result, recurrent networks are commonly employed for unsupervised learning tasks.

Recurrent neural networks, like their feed-forward counterparts, often consist of a visible input and output layer and intermediate/hidden layers that learn to model time-dependence. In addition, RNNs share the same weight parameters within each layer, while feed-forward models have different weights across each node. Nevertheless, both architectures use backpropagation and gradient descent to determine the parameters of the model that are optimal. RNN utilises backpropagation through time (BPTT) as a means of adjusting for the self-connections. Though conceptually similar in the sense that both approaches adjust model parameters in accordance with some error signal at the output layer, the BPTT variation differs somewhat. Here, the error is summed up at each time step, where by comparison feed-forward networks do not require this error summation since there are no shared parameters across each layers. Nevertheless, this can lead to problems such as vanishing and exploding gradients, as well as complicating computation for long-term time series. If the gradient determining the updates to the parameters becomes too small (vanishing gradient), the weight updates become insignificant and no learning occurs. Conversely, too large (exploding) gradients can result in unstable updates and poor performance.

To address the vanishing gradient problem for application domains where it is beneficial to retain information over time, long-short-term memory (LSTM) networks have gained popularity (Hochreiter and Schmidhuber, 1997). In addition to the standard RNN units, LSTM use special units called memory cells that can maintain information in memory for long sequences and learn to select stored information from the earlier portion of the sequence, if needed.

2.2.4 Transformer

Recent developments have seen a significant increase in the use of transformer networks in various domains. This architecture was originally proposed for Natural Language Processing, where it has proven superior to both LSTMs and RNNs in a variety of benchmarks. Computer vision has recently shifted towards this architecture as well.

The primary objective of these models is to handle large quantities of training data. Using large models that have been pre-trained using a significant amount of unannotated data, the goal is to provide a general-purpose deep learning model that can be adapted to downstream tasks.

Transformers, as originally proposed by Vaswani et al. (2017), operate on an encoder-decoder architecture in which the encoder processes the input into an encoding that contains the most relevant information from the input. By decoding the encodings and incorporating the associated information, the decoder generates the output. This principle has been extensively utilised in autoencoders, where a feed-forward neural network is employed to compress the input into an intermediate layer of lower dimensionality with an output layer that has the same number of nodes (neurons) as the input layer. In the transformer architecture by Vaswani et al. (2017), both the encoder and decoder are composed of a feed-forward neural network and a self-attention mechanism. In addition to the original neural machine translation task, transformers have been adapted to a variety of other applications, in which the decoder component may not be necessary since no data is generated.

The novel component is a mechanism referred to as attention which, which performs different weightings on the various parts of input data. It was originally intended to model sequences and the relations between them without using recurrent units (Vaswani et al., 2017). A powerful feature of the architecture is its ability to capture relational information. In particular, the so-called multi-head self-attention mechanism is responsible for the effectiveness of this approach.

In a neural network, attention is a mechanism by which a model can learn to predict by selectively focusing on a given set of data. The amount of attention is measured with learned weights, and the result is usually expressed as a weighted average. In self-attention, the model makes predictions about one part of an observation related to the same sample based on other parts of the observation.

In the self-attention mechanism, input is encoded as sets of key-value pairs K and V. Similarly, the query $\mathbf{Q}$ is intended to facilitate a search mechanism for relevant information. Formally, we define $\mathbf{Q} \in \mathbb{R}^{L \times d_k}$ as query matrix, $\mathbf{K} \in \mathbb{R}^{L \times d_k}$ as key matrix, and $\mathbf{V} \in \mathbb{R}^{L \times d_v}$ as value matrix, where L is the length of the input sequence. d_k and d_v correspond to the dimensions of the query and the value vectors, respectively.

The sum of the value vectors is weighted, while the weights assigned to the values are determined by the product of the query matrix with the keys:

$$\text{Attention}\left(\mathbf{Q}, \mathbf{K}, \mathbf{V}\right) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}\right)\mathbf{V}$$

In the current transformer implementations, a slight modification is made to include additional key, value, and query matrices.The reason for this modification stems from the fact that regular dot-product attention is prone to attending mostly similar words since their embeddings will be similar. In order to learn more complex relations, this additional step of mapping the translation into additional key, value, and queries provides a way to circumvent this limitation.

A multi-head mechanism distributes the inputs into smaller chunks and then computes the scaled dot-product attention over each subspace in parallel, rather than computing it all at once. Calculating the scaled dot-products over subspaces in parallel allows the so-called multi-headed attention to be calculated, further increasing the complexity. The independent attention outputs are then concatenated and transformed into the dimensions required. It is intended to “jointly attend to information from different representation sub-spaces at different positions” (Vaswani et al., 2017).

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \left[\text{head}_1; \ldots; \text{head}_h\right]\mathbf{W}^O$$

where

$$\text{head}_i = \text{Attention}\left(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V\right)$$

and $\mathbf{W}^O, \mathbf{W}_i^Q, \mathbf{W}_i^K$, and $\mathbf{W}_i^V$, are learneable parameters, where $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$ and $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ (d_{model} being the embedding size).

In addition to its ability to handle the complexity of tasks such as NLP, this architecture has also exceeded expectations within the domain of computer vision (Radford et al., 2021) owing to the amount of information it can process through the large number of parameters it contains.

2.3 Why Explainable AI is Challenging

Although recent advances in artificial intelligence and machine learning have contributed to further complexity of the models, it is not a new problem that lack of transparency in these systems is a concern. Starting from the late 1970s, expert systems have been created in order to achieve human-like reasoning capabilities across a variety of application areas. Explanations of the inner workings of these systems were desirable not only since they could be used to improve the development of these systems, but also to promote trust between users and the systems' outputs. Contrary to today's connectionist AI and ML systems, expert systems were computationally symbolic in nature. In theory, explicit symbolic representations rather than their distributed connectionist counterparts would permit easier access to individual reasoning steps and chains of inference. Yet, in many instances the symbolic traces of a system's decision were also too complex for direct interpretation (see, e.g., (Preece et al., 2018) for a modern example).

A central feature of deep learning and most connectionist systems is the process of learning that takes place during the training phase. Unfortunately, this process can potentially, via its learning outcome, also lead to unanticipated behaviour of the system. As we hitherto have no standard method for testing the decision logic the system inferred from the training data, it is challenging to anticipate undesired system behaviour before it occurs: The rules governing the system's operations are inducted purely from observation of the training data which establishes the system's internal connectionist representations. Current machine learning algorithms utilise the large amounts of data and computing power available for training in order to identify complex relations between input features; however, disaggregating these complex relationships into an understandable manner is difficult, which ultimately complicates or altogether prohibits oversight and control at scale. Consider, as an example, the fact that learning from historical data may lead to the propagation of established misconceptions. Often the underlying rules are obscured within the optimised model, possibly resulting in poor generalisation and perpetuation of undesirable practices and prejudices.

State of the art machine learning methods (especially Deep Learning) are black boxes that make them unintelligible. In this section, we outline intuitively the technical properties that contribute to the opacity and why this makes explaining what exactly is occurring in these models challenging. As a starting point, we describe what a classification model does, regardless of whether it is highly parameterised and complex, or sparse and simple. We consider a set of training data distributed across the input space. As individual data points have labels or classes assigned to them, the model is applied to identify the underlying feature representations that enable us to distinguish between different classes (i.e. to identify an encoding of data properties that ultimately enables the model to distinguish between classes). One of the reasons for the lack of interpretability of the model is that the feature representations are not known but learned during the optimisation process. Moreover, complex connectionist models employ distributed representations, which means that even if we were to extract the learned representation, individual data points would be encoded as high-dimensional vector representations of the features. This is depicted in the following illustrations, which highlights some of the key challenges in this regard.

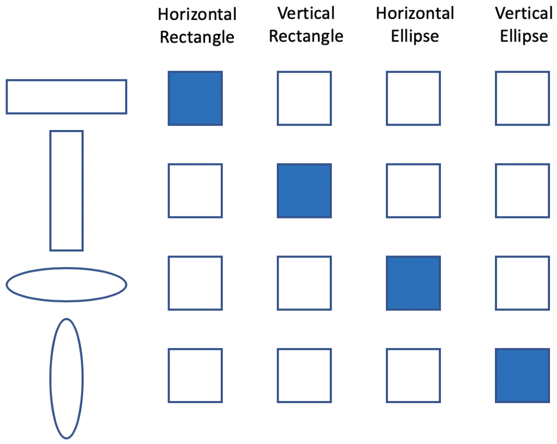


Figure 2.1: Illustration of a local representation (inspired by Hoffman (2018)).

Figure 2.1 illustrates an example of a non-distributed representation that is sometimes described as a "local representation" or "sparse representation". The dimensionality of this type of representation, for instance, would increase as the number of inputs increased, so it is considered rather inefficient. Aside from this, the representation does not include any information about how features relate to one another. The principle strength of distributed representations is that they are capable of accounting for "semantic similarities" between data points and their features. Figure 2.2 provides

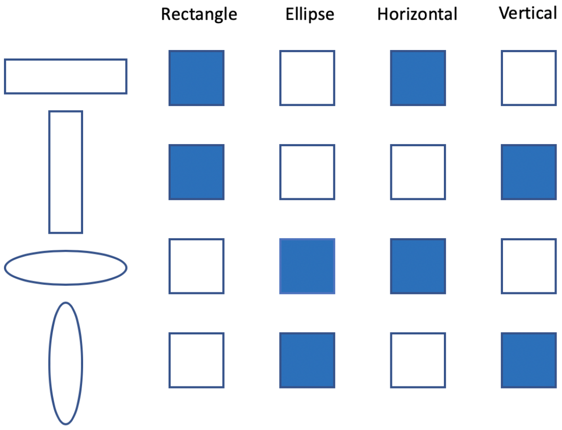


Figure 2.2: Illustration of a distributed representation (inspired by Hoffman (2018)). In a Neural Network, the distributed feature representations are learned during the training phase.

an illustration of a distributed representation of the same data. As multiple features now jointly represent the concepts within the data, the efficiency of information representation within these data-points has been greatly improved. Information regarding relationships between concepts can be obtained, e.g., from joint occurrences of certain individual features. Relationships between data points can be identified, for example, by identifying the joint occurrence of concepts using several individual characteristics. A substantial amount of research has been invested in discovering meaningful computational features that describe data and the relationships between data points while also allowing for good predictive performance of models based on these representations (Arel et al., 2010). An important factor contributing to the current success of deep learning is precisely the ability to employ training algorithms to learn rich distributed representations automatically from data. Developing a method to translate the distributed representation and operations into understandable information is a significant challenge.

2.4 Approaches to Explainable AI

It is the purpose of this thesis to present various methods that attempt to make sense of these distributed representations. Our next two subsections will introduce two approaches that we will use in order to conduct a variety of experiments.

2.4.1 Lime

As a first method we introduce a local surrogate approach, in which interpretable models are used to explain individual predictions of the underlying black-box model. Interpretable model-agnostic local explanations (LIME) refers to an approach in which variations in the input data surrounding the instance of interest are used to probe a black-box model Ribeiro et al. (2016). Using perturbed samples and the corresponding model predictions, LIME generates new data. Using these generated data points, an interpretable model is developed on the basis of weighting the samples by their proximity to the instance prediction that is being explained. Typically, the interpretable model of choice is a simple linear regression, although other models may be applied. Although the method was originally developed for tabular data, it has since been adapted for other types of data. By using sections of an image that have been grayed out, pertubations are created in images. By doing so, the interpretable model can learn which areas are important for the black-box model to predict the output. Local surrogates can be described in the following manner Ribeiro et al. (2016):

$$\text{explanation }(x) = \arg\min_{g \in G} L\left(f, g, \pi_x\right) + \Omega(g)$$

where the explanation for a given instance x is model g which minimises a loss L measuring how close the explanation is to the black-box prediction of model f, while maintaining a low level of model complexity $\Omega(g)$. G refers to the family of possible explanations and π_x defines the size of the neighbourhood around the instance x that is explained. We refer the reader to Ribeiro et al. (2016) for a more comprehensive description of the entire approach.

2.4.2 Shapley

An alternative local explanation method that has recently gained popularity is the Shapley value approach. Originally proposed by Shapley (1953), it has recently been adapted for machine learning models by Štrumbelj and Kononenko (2014) and Lundberg and Lee (2017). Essentially, the goal is to capture the average marginal contribution of each feature value across various possible feature combinations. The Shapley value for a particular feature of a specific input represents its contribution to the prediction with respect to the average prediction for the dataset Lundberg and Lee (2017). To determine this value, the authors propose that features are systematically added to the model to determine the average change in prediction. The SHAP method is applicable to both classification and regression tasks. In combining various feature attribution approaches into one framework and publishing an accessible implementation, Lundberg and Lee (2017) contributed to a significant increase in the popularity of this methodology. According to SHAP, an explanation is provided as follows:

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

where g is the explanation model, $z' \in 0, 1$ is referred to as coalition vector, M is the maximum coalition size and $\phi_j \in \mathbb{R}$is the feature attribution for a feature j, also known as Shapley values. With a value of 1, the coalition vector indicates that a feature is present, whereas a value of 0 signifies that it is absent. In order to compute Shapley values, we simulate that only some features are in play ("present") while others are not ("absent"). This representation of coalitions allows the computation of Shapley values ϕ by employing a simple linear model. In this representation, the instance of interest x translates to a coalition vector of all 1s, with all value features being present. It is pertinent to note that in the vast majority of ML models, a feature cannot be absent. The SHAP method computes outputs for these combinations by replacing absent features with sampled feature values from the dataset and averaging the results. As a result, Shapley values can be computed for all coalitions. A more detailed explanation with examples can be found in Lundberg and Lee (2017) and Molnar (2019).

Chapter 3

Taxonomy of XAI methods

In the following chapter, we intend to outline the current landscape of research on explainable artificial intelligence in a systematic manner. One of our goals is to provide a practical framework that will extend previous survey on rule-extraction methods that were considered outside of the recent surge of interest in XAI. In order to accomplish this, we extend and adapt an existing taxonomy by Andrews et al. (1995) in accordance with current trends. We can use this as an opportunity to form a framework that not only analyses recent developments, but also systematically encompasses a wide range of work that can be considered Neural-Symbolic. This is necessary in order to align Neural-Symbolic approaches which provide some XAI characteristics with existing XAI research. By conducting this systematic review along the framework dimension introduced earlier, we are also able to gain a deeper understanding of the proposed methods and their differences. By doing so, we will be able to formulate an overview table along the proposed dimensions. We will then use this overview to stimulate further investigation of specific promising approaches that address the XAI challenge from a neural-symbolic perspective, as well as incorporate emerging trends that will meet the current XAI demands and identify gaps.

In light of the large variety of interpretability methods that are employed in the surveyed publications, we identified the need for a deeper investigation of comparisons between the different approaches. The proliferation of explainability papers tends to make it increasingly hard to find comparable methods aiming to gain insight into the behaviour of a Machine Learning model, particularly as new approaches are proposed. An effective comparison of methods would require authors to engage actively with the following criteria. Moreover, a better understanding of the approach should be accessible more readily without requiring extensive disentanglement of technical details.

In the survey by Andrews et al. (1995) (henceforth, ADT), the authors introduce a taxonomy allowing different rule extraction techniques to be categorised, discusses how they operate, and defines criteria for evaluating their effectiveness. The authors develop a schema for the extraction of rules from trained neural networks based on a coherent taxonomy. The first five dimensions have been derived and extended from the existing taxonomy in order to apply them to the broader field of XAI beyond rule extraction. It follows that while the ADT dimensions are related to the first five dimensions, they are not identical. The reason for this is that they were intended to describe methods for extracting rules, which are only a subset of the methods we outline below. This is done by extrapolating the ADT dimensions, while preserving the original intention. The three additional dimensions (6-8 in the list below) are presented because they could be considered the most descriptive of the current XAI approaches. While these three dimensions have been discussed before to describe XAI methods, the combination and integration into the taxonomy is novel. The proposed dimensions are:

- Quality of the information representation
- Expressive power of the information representation
- Translucency of the model
- Portability of the method
- Complexity of the algorithm
- Locality of the explanation representation
- Stage of the method application
- Use cases of the approach

Below, we will discuss the taxonomy and its dimensions. Many of these dimensions and this taxonomy are applicable to a wide selection of XAI methods.

1. The quality of the extracted information can take many different forms. Accuracy and fidelity are two of the most relevant metrics. Typically, the accuracy of an explanation representation may be determined based on a dataset in accordance with the standard measure of model performance. *Fidelity* describes how accurately the information represents the model that the method is attempting to explain or interpret. It can be calculated using any dataset to measure the percentage of predictions that are congruent between the explanation and the

original model. Consequently, the explanation may yield 100% fidelity while achieving less than 100% accuracy, when the extracted representation mimics all misclassifications of the main model.

For both of these metrics, it is assumed that the representation used to comprehend the behaviour of the complex model is capable of data inference from raw input to the desired output space and is therefore able to perform the task independently of the underlying model. As such, this holds true for models that are inherently interpretable, as well as for example extracted rules, trained surrogate models, and distillation methods. In cases where the method is limited in its ability to establish input to output mappings, particular adaptations such as local fidelity may be required. However, if this is not the case and the representation provides only excerpts or summative information from the model, the *quality* of the information representation refers to the truthfulness or soundness of the explanation, more specifically to how closely the simplified representation matches the true inner working of the model.

2. The expressive power of the derived representation refers to the breadth of information that is reported. Originating from symbolic rule extraction, where for example boolean and fuzzy logic are distinguished, it is also applicable to the various types of explainability methods proposed more recently. In this context, there are a number of possibilities for presenting information, such as coefficients, heatmaps, or more general measures of feature importance. The type of representation is usually drawn from the task and its corresponding data domain.

   Additionally, this taxonomy dimension is closely related to the level of abstraction of the explanation, which can result in different levels of intuitiveness that makes the information itself more accessible and more readily understandable. It is possible that different requirements will apply depending on the context and the time available to revise and digest the information. Moreover, a consideration of the target audience and the expertise of the individual is imperative. Data scientists, compliance officers, decision-makers, and other stakeholders may have different backgrounds and different areas of expertise as well as experience in the topic to which the model is applied. Thus, the nature of the explanation and its expressive power depends on the consideration of the recipient of the explanation.

3. *Translucency* refers to the degree of granularity at which we approximate the behaviour of the models. Generally speaking, one may consider translucency as the depth of the explanation and the extent to which one wishes to represent the reasoning and functioning of the internal behaviour of the represented model. To accomplish this, it is necessary to incorporate the model's latent representations and their responses to different inputs. Contrary to this, many methods are aimed at mimicking the input-output behaviour in a comprehensible manner. There are three levels of detail derived from the Neural Network rule extraction methods (Andrews et al., 1995). First, so-called pedagogical approaches, where models are being treated as black boxes. Second, decompositional approaches, which builds the explanation out of a multitude of smaller components of the explained model, keeping in mind the internal representations of the model. And third, eclectic approaches being hybrids of the two methods. In spite of the fact that these definitions were derived from neural network rule extraction, they remain relevant and applicable to most post-hoc explanation techniques.

4. *Portability* of the approach describes the architectural and training requirements of the model we are seeking to explain. As a best case scenario, there are no special requirements and the method is model-agnostic and therefore applicable to all underlying models.
   For example, the portability dimension may be closely related to the degree to which a method can be tailored to particular previously trained models. If the objective of an XAI method is to provide a sound and truthful representation of a model, this can often only be accomplished by taking advantage of the inherent biases of the underlying model. Nevertheless, this limits the applicability of the method to this particular model architecture. Following from the dimension of translucency above, pedagogical approaches are typically model-agnostic as they only consider input and output, whereas decompositional methods also focus on the inner representations and workings of the approximated model resulting in constraints.

5. *Complexity* describes the effectiveness and efficiency of the search for a comprehensible representation. Even though an exhaustive search could provide rules for the entire input-output behaviour of the original model, such a search would often be computationally infeasible. Although this trade-off has its roots in the

search for logical rules, it also applies to non-symbolic methods. There is a trade-off between the truthfulness of the explanation and the complexity of the search underlying many approaches. In this regard, permutation-based methods are an intuitive example, which are typically limited to isolated features, since the computational resource required would not permit their extension to feature interaction. Here, permutation-based approaches refer to the process of manipulating input data in a systematic manner while simultaneously observing changes in the output to discern patterns of behaviour. The more interaction we wish to accommodate, the more computationally expensive the process becomes.

6. *Locality* indicates how many instances are to be considered in an explanation, and thus, describes a specific level of abstraction of the explanation. Generally, we distinguish between two levels, namely local and global. A local explanation is only applicable to a single instance, hence, one input-output pair. On the other hand, global explanations are generalizable across all instances of the trained model.
   A distinction between these two paradigms of explanation is essential as they are fundamentally different. A global explanation describes the model as a whole, whereas a local explanation describes one particular instance.

7. *Stage* of the method application describes the process stage in which the method is applied. It may be divided into three distinct categories (based on the presentation by Khaleghi (2019)). Each dimension will be further explained in section 3.1.1.
   A pre-training approach is intended to provide insights into the data and its implications for the model prior to the optimisation based on data. In particular, these methods can be used to identify unrepresented classes in the data in order to detect unsound grounds for a model to be effective.
   Inherently interpretable models provide ways to understand the behaviour of the models by introducing particular constraints. An example of such constraints may be requiring a simpler model structure or enforcing monotonicity.
   Lastly, Post-Training methods (also referred to as post-hoc), which require the use of a trained model after it has been optimised for a particular task, aim to gain a deeper understanding of the model's behaviour on a particular input instance or globally.

8. The *use-case* of the method relates to the intended data domain for the approach. Although many methods are typically designed to excel in particular data domains, they are often adapted to address alternative data domains as well.

3.1 Characterisation of Methods

There is an absence of coherent structure in the field of explainable Artificial Intelligence, as there is no standard procedure to differentiate between methods, which may lead to a great deal of confusion in regards to what methods are appropriate and what alternatives are comparable. To make the allocation of individual methods intuitively clear, a suitable structure should permit easy application without extensive prior knowledge of the entirety of the field. Today, the most common method of distinguishing between approaches is to indicate its locality. As a result, all methods are restricted to two categories, namely global and local interpretation. It is, however, difficult to find similar approaches due to the variety of methods available to achieve local or global interpretability. The following framework presents a practical approach to achieving local or global interpretability. Instantiation has been chosen as the guiding dimension due to its popularity and intuitive nature in the data science process flow. Therefore, practitioners may assign desired methods in accordance with their own constraints and requirements.

1. Pre-Training

2. Training Inherently Interpretable Models

3. Post-Training

Although we will discuss methods in all three categories to establish the taxonomy, our main focus will be on the post-training category as the majority of XAI methods as well as our proposed method fall into this group.
Furthermore, it is important to note that different approaches and techniques are needed for different tasks. In particular, different data types pose different challenges when it comes to explaining an instance or model. Classification models for images could be better explained through visual representations, whereas tabular data could perhaps be better explained through rules, feature importance, and probabilities. Such specificity also applies to natural language or time series data. Nevertheless, many methods can be applied in different contexts. When not explicitly stated,

described techniques have been applied to various tasks or could be applied to various tasks.

3.1.1 Pre-Training

Pre-training methods are typically used before developing a model, since it is vital to analyse and understand the data before selecting the appropriate model. Therefore, pre-training explainability techniques are independent of a model as they are intended to be applied to the data themselves (Arya et al., 2019).
It is important to note that datasets are becoming increasingly large and complex. As the number of features and data entries increases, analysis becomes more challenging, as traditional logic extraction and visualisation techniques have limitations with regard to scalability. The following sections will present some representative methods that may be adequate to handle current practices and dataset sizes. There are several approaches that facilitate data interpretation, such as the obtention of meaningful representative features and sparsity (a small number of features) (Carvalho et al., 2019). Pre-training interpretability is therefore closely related to data interpretability, which is realised by exploratory data analysis techniques. Among them are classical descriptive statistics, as well as data visualisation methods, such as Principal Component Analysis (PCA) and t-SNE (t-Distributed Stochastic Neighbor Embeddings), as well as clustering methods, such as k-means and MMD-critic (Maximum Mean Discrepancy). All of which will be briefly outlined below. Using these approaches, data visualisation can be accomplished, which in turn becomes crucial to the interpretation of data before the model can be built. It consists of the visual presentation of information and data in order to facilitate a better understanding of the data and resulting implications for the model. Thus, determining whether the model's assertions are reliable requires that one understands the underlying data structure and relationship.

Among the most prevalent methods is the practice of looking for prototypes and criticisms. Prototypes are representations of the underlying data distribution. By contrast, a criticism refers to an instance which is not represented by such prototypes, so it is an outlier. Many approaches have been developed to identify prototypes within data sets. Clustering algorithms that return data points of the dataset as cluster centres are appropriate for finding prototypes. Among these clustering algorithms is the k-medoids method which is based on k-means (Kaufman and Rousseeuw, 2008). This method aims to minimise the distance between a point designated as the centre of a cluster and the points that are labelled as being in the cluster. With k-medoids, actual data points are selected as centres, allowing for greater interpretability than with k-means, wherein the centre of a cluster is not necessarily one of the input data points.

Many clustering methods, however, are primarily concerned with finding representative points as opposed to outliers, such as criticisms. As such, the MMD-critic introduced by Kim et al. (2016) focus on the discovery of criticisms in the same manner as prototypes. By comparing the distribution of the data with the distribution of selected prototypes, the algorithm identifies prototypes which minimise the disparity between the two distributions. The instances in densely populated areas are considered prototypes, whereas examples outside these clusters which are drawn from the data distribution are considered criticisms.

We can utilise visualisation methods to identify key features and meaningful representations from our data that indicate what might influence the way a model takes decisions in a human-comprehensible manner. Since we are often dealing with a vast feature space, dimension reduction techniques are important since it helps to reduce the feature space and, in turn, give us more compact representations that can be visualised, and thus show us what might be influencing a model to make a certain decision. In order to enable visual interpretation, the following representational techniques are commonly employed.

One of the most widely used techniques is PCA (Pearson, 1901; Jolliffe, 2002). This approach aims to reduce the feature dimensions of a dataset while preserving as much variability as possible. Specifically, this is accomplished by identifying new variables that are linear functions of those in the original dataset. Each of these linear functions maximises the variance of the data and is uncorrelated with the other. These so-called Principal Components (PCs) represent the number of uncorrelated features in which the first PC is the source of the greatest amount of variance in the data, followed by the second PC, in decreasing order. Despite its robustness, it does have some limitations. A PCA may fail to identify the most compact description of data if the principal components are statistically dependent yet uncorrelated (Jolliffe, 2002).

Self-organising map (SOM) is another popular method for reducing the dimension of the data while maintaining its topological structure (Kohonen, 1982). SOM is characterised by its ability to generate effectively spatially organised internal representations of the input signals of features and their abstractions and is a neural network architecture originally intended for unsupervised machine learning.

Latent semantic analysis (LSA) is a method that was developed to improve the effectiveness of information retrieval systems (Landauer et al., 1998). Dimensionality reduction is achieved through the creation of clusters of data that are based on the latent representation. As a result, the semantic similarity contained within the latent vector representation is utilised for comparing data points and representing data in a lower-dimensional space.

The t-Distributed Stochastic Neighbour Embedding (t-SNE) (Van Der Maaten and Hinton, 2008) is a nonlinear dimensionality reduction technique that is suitable for embedding high-dimensional data in a two- or three-dimensional space to facilitate visual analysis. In particular, each high-dimensional object is represented as a low-dimensional point in such a way that similar objects are modelled as nearby points and dissimilar objects are modelled as distant points. Initially, t-SNE develops a probability distribution over pairs of high-dimensional objects by assigning a higher probability to similar objects, whereas a lower probability is assigned to points that are dissimilar. Then, the t-SNE method establishes an equivalent distribution over the low-dimensional points, and minimises the comparison of the Kullback-Leibler divergence between the two distributions.

An automated generative approach using variational autoencoders (VAE) by Kingma and Welling (2014) refers to a type of artificial neural network that can be used to learn efficient encoding methods for unlabelled data. Validation and refinement of the encoding takes place by attempting to reconstruct the input from the encoding. By training the network to ignore noise in the data, autoencoders learn how to represent a set of data more efficiently.

One illustration is the application of identifying a molecule and its governing factor interactions in genomics research. There are however thousands or even millions of features within these datasets that are active in a variety of cellular and developmental contexts. Interpreting and comprehending the data at this magnitude is a challenge. Therefore, practitioners often apply sparsity penalties or dimensionality reduction techniques to make the data manageable and comprehensible to facilitate the identification of promising candidates for experiments (Murdoch et al., 2019).

The use of word embeddings and t-SNE to visualise the semantic similarity of text features before training a model is another example of insights obtained before the model optimisation process (Molnar, 2019).

To summarise, pre-training explainability refers to a variety of methodologies employed to gain a better understanding of the available dataset prior to modelling. Using pre-modelling explainability to extract data-related insights can help to build AI systems that are more efficient, understandable, and robust. Using methods such as t-SNE or PCA to reduce the dimensionality of the data allows us to remove noise from the data that hinders its comprehension. The reduction in complexity, however, may not be desirable and appropriate in all circumstances. Some low-level representations, such as an image, cannot easily be translated into a lower dimensional representation. In addition, it is precisely this complexity and weak regularities that black-box models often rely upon to achieve high performance. Therefore, the elimination of weaker regularities could lead to diminished effectiveness in particular domains. Moreover, these methods are fundamentally limited by the fact that they do not account for the variety of ways in which the model may learn undesirable behaviour during the optimisation stage. Consequently, these approaches are most effective when combined with alternative explainability methods that emphasize explaining what has been learned.

3.1.2 Inherently Interpretable Architecture

Whenever a machine learning system is employed on a task, it may be advisable to determine to what extent comprehensibility is an essential criterion. In the event that comprehensibility is of critical importance to the project's outcome, it is imperative to develop a model based on interpretable techniques from the outset. The majority of approaches in this category can be interpreted globally due to the fact that these types of methods incorporate architectural features resulting in some level of interpretability.

The architecture of the model can be restricted in a way that makes its behaviour more understandable, thus achieving interpretability inherently. Among the methods for accomplishing this is through the imposition of linearity between features and targets. Moreover, monotonicity can be enforced, which ensures that the relationship between a feature and the outcome is consistent, either increasing or decreasing, over all values in the feature range. Further differences in methods and complexity can be found in the way feature interactions are dealt with. When feature interactions are taken into account, models usually perform better but their complexity also increases, hence affecting their interpretability. As an example, Decision Trees are a method allowing for nonlinearity and complex interaction of features, while still maintaining the ability to follow their behaviour. However, an arbitrarily large number of feature interactions associated with a deep tree may limit the interpretability of the system. We present interpretable models that are rule-based, which are comprehensible global predictors that successfully execute decisions by describing the mapping as a set of rules. These methods are most appropriate for use with tabular data in which the features themselves are meaningful in isolation.

Letham et al. (2015) present a model that is described as being inherently concise and convincing to domain experts. The model is implemented as a Bayesian Rule List based on Decision Trees. It calculates a posterior distribution over possible decision lists, thereby assigning probabilities to potential splits. The confined architecture of this sparse Bayesian generative model has demonstrated less predictive performance than an unconstrained generative model, as would be expected.

Lakkaraju et al. (2016) introduce a framework for constructing models that aim at obtaining interpretable and precise predictions in which Interpretable Decision Sets (IDS) are extracted. Therefore, the expressive power is limited to propositional if-then rules. In this approach, each rule is independently applicable and the resulting decision sets are deemed to be simple, and easily understood. Furthermore, this approach aims to discover accurate, concise, and non-overlapping rules that cover the entire feature space. It is based upon the belief that decision sets are more efficient than decision lists. The rules in decision lists are implicitly dependent on all prior rules, while the rules in sets are independent of each other.

In Malioutov et al. (2017), the authors propose a linear programming approach to infer both conjunctive and disjunctive clause rules based on training data. The optimisation formulation allows the resulting rule-based global predictor, known as 1Rule, to automatically adjust accuracy and interpretability. As opposed to other methods, this approach is less dependent on heuristic solutions, but computationally more costly.

The methods above have significant advantages when the task requires simulatability. Simulatable models are those in which a human can simulate and reason about the decision-making process, anticipating the possible outcomes. As a result, we may predict the output of a trained model for an arbitrary input. Because of their human readability, rule-based methods have a very high degree of simulatability. These methods can be very effective provided that they are also capable of delivering an appropriate predictive performance. An example of a real-world deployment of these methods is in hospitals. When patients are diagnosed with atrial fibrillation, patients, relatives, and caregivers are often interested in predicting the chances that the patient will suffer a stroke in the coming year. Moreover, given the potential repercussions of medical decisions, it is vital that predictions be not only accurate, but also easily understood by both caregivers and patients. By virtue of their inherent transparency, rule-based methods facilitate the anticipation of and trust in decisions (Kaufman and Rousseeuw, 2008).

Among the interpretable models, they may be the most intuitive. In order for this statement to be valid, the number of rules must be small, the conditions must be short, and the rules must be organised into a decision list or non-overlapping decision set. Decision rules generated by the methods can be as expressive as a decision tree, yet more compact. Many decision trees also suffer from replicated subtrees, which occurs when both a left and a right child node have the same structural elements. Due to the threshold in the conditions, decision rules provide robustness against monotonic transformations of input features. These are also robust against outliers, as it is only necessary to determine if a condition holds or not.

However, rule-based methods are typically ineffective for describing linear relationships between features and outputs. They share this limitation with decision trees. Neither decision trees nor rules are capable of producing smooth curves for predicting outcomes, but only discrete steps. The issue here relates to the fact that it is the most applicable for categorical inputs. Prior to making use of numerical features, it is necessary to categorize them. While it is possible to cut a continuous feature into intervals in a variety of ways, it is not simple to achieve and involves many challenges.

Almost all of the research and literature in the field of rules focuses on classification and frequently overlooks regression. Although a continuous target can always be segmented into intervals and transformed into a classification problem, you will always lose some information. It is generally more attractive to employ an approach that can be applied both to regression and to classification problems.

The following set of methods is dedicated to approaches returning an intelligible predictor equipped with a comprehensible explanation using representatives. A prototype is a representative for a set of similar instances. Having prototypes among the observed points is desirable for descriptive accuracy, but often results in inferior predictive accuracy.

Kim et al. (2014) present the Bayesian Case Model (BCM), which involves a comprehensible predictor that can learn prototypes through clustering the data and identifying distinct subspaces. Each individual representative of a cluster is selected and labelled a prototype. In order to decompose a space into subspaces, a set of features are taken into account that are crucial to distinguishing clusters. Prototypes accompanied by descriptions of their essential characteristics are presented in the explanations. As with all Bayesian approaches, choosing the correct prior can be challenging in different situations. Additionally, the approach can become computationally intensive when a multitude of features are present.

Bien and Tibshirani (2011) present the transparent Prototype Selection (PS) method for selecting prototypes in a classification setting. It is based on the principle that finding representative samples from a large data set has greater interpretative value in certain domains than generating an optimal linear combination of elements within the data set. Every prototype is a component of the real data, for example, actual hand-drawn digits. As an example, in medical applications, this would mean that prototypes correspond to actual patients, providing useful information to domain experts who want to interpret the underlying data sets. This method can be used solely for finding prototypes, however the authors propose that it be used as a classifier as well, by applying the nearest-neighbour rule to the set of prototypes.

Models based on prototypes have been applied to a variety of data types, but predominantly images and tabular data. An interpretable and sparse data representation is generally considered truthful if the prototypes are representative. This method is best suited to explaining general behaviour without explicitly requiring the explanation to comprehend the entire operation of the model on all instances within the dataset. Kim et al. (2014) demonstrate the ability to gain insight into image recognition by examining representative prototypes. Bien and Tibshirani (2011) describe their method and its utility in the biomedical field, specifically protein classification.

The methods to follow describe a set of approaches to the problem of explainability that are based on inherently comprehensible architectures but are distinct from the set of methods previously examined. Caruana et al. (2015) present a generalised additive model (GAM) learning method that can be adapted for different applications. In spite of the author's focus on pneumonia, the method is being applied to a wide range of other medical applications (Caruana et al., 2015). In their paper, GAMs are introduced as the benchmark for comprehensibility when solely univariate terms are considered. The explanations describe the contribution of different features as well as their shape functions, which provide insight into a feature's linearity. When taken together, this allows for a comprehensive review of different models. In their work, the authors primarily focus on the application of this method to tabular data.

The authors of the method apply this model to the problem of patient prioritisation for pneumonia patients in a hospital. Using this method, the authors research the problem of determining which patients in a hospital should be given priority treatment for pneumonia. A model was developed to predict the likelihood of death within the next 60 days and patients with a greater mortality risk were granted a higher priority. As a result of a thorough examination of the univariate and pairwise terms that were interpreted as curves and heatmaps, researchers found various counter-intuitive properties indicative of faulty model behaviour. A particular confounding factor, a specific preconditions, was picked up by the model as a positive signal because these patients had previously received more examination & intensive treatment leading to their recovery. As a result, researchers were able to identify and correct for errors using the interpretable model, thus ensuring that the model can be trusted in practical application.

Recently, this approach was extended to include neural networks. In Agarwal et al. (2021), the authors integrate multiple neural networks in place of linear models to reach the same feature-based explanations. Using this approach, the model learns to perform tasks by combining a linear combination of neural networks that each pay attention to a specific input feature. In terms of accuracy, they are comparable to existing generalised additive models, however, since they are based on neural networks rather than boosted trees, they are more flexible. The increase in the number of learnable parameters, however, also causes an increase in the computational cost.

GAM, GLM, and their extension have been around for many years, and therefore have been extensively researched. Particularly, the wide range of extensions provides a flexible set of tools that can be used across a variety of fields. In spite of this, the number of ways to extend the simple linear model can be overwhelming. Furthermore, most modifications to the linear model render the model less interpretable.

Ustun and Rudin (2016) propose a sparse linear model called SLIM that is used to create scoring systems based on the data collected. It is motivated by the controversial scoring system for criminals called COMPASS (Larson et al., 2016). Based on the high level of sparsity and the small number of coefficients, it provides interpretable capabilities that enable qualitative understanding of predictions. With the application of SLIM to the COMPASS dataset, it is possible to achieve the same predictive accuracy while maintaining the full descriptive power. Consequently, they were able to detect any bias and illegal discrimination that may have been present. While this sparsity in the model is adequate for the task they are presenting and allows for straightforward interpretation, it lacks the effectiveness to solve tasks in more complex scenarios.

Wu et al. (2018) propose the use of a model complexity function in order to find more interpretable neural networks during the training phase. Essentially, the penalty favours models that have decision boundaries that can be easily approximated by small decision trees and is therefore referred to as tree-regularisation. The average length of a decision path is defined as the human simulatability factor. A main contribution of this work is the use of a surrogate regularisation function while training and mapping each candidate neural model parameter vector to an estimate of the average path length. A multilayer perceptron network is used to implement the regularised approximation function and thus achieve differentiability. A discussion of three different medical applications is presented, namely predicting therapeutic outcomes of hospitalised septic patients, predicting HIV treatment outcomes, and human speech processing. The authors argue that the decision trees which mimic the predictions of a constraint tree-regularised deep model are small enough to be easily simulated by the user while imparting to the practitioner an understanding of the model's inherent nonlinear predictive behaviour (Wu et al., 2018).

The boundary between inherently interpretable methods and post-training methods is in some instances fluid. In some approaches, the system generates an inherently interpretable model in the form of a sparser model that attempts to replicate the behaviour of an unconstrained black box model.

3.1.3 Post-Training

In some applications of machine learning models, accuracy is of paramount importance, but a certain level of comprehensibility is also desired. In some circumstances, the structure and shape of the input data hinder the use of highly interpretable models such as linear regression, despite the necessity for an understandable outcome and transparent behaviour of the model. In the following methods the approaches aim to explain the behaviour of complex, nonlinear, non-transparent, black-box settings by approximation. Some of these techniques are not limited to providing insights into black-box models but can also contribute to enhancing the understandability of partially interpretable models.

Following the training of a model, there are several ways in which insights may be gained. In some cases, a statistical test is employed, while others aim to obtain logic information from the models. Throughout this process, we will review a selection of representative methods across a wide range of approaches.

The term “feature importance” refers to the extent to which a predictive model is dependent on a distinct feature. The assigned importance scores provide information about the degree to which a particular feature contributes to a particular decision or the model's decision making process more generally. However, the latter is frequently an aggregate of the former. In an intuitive sense, features that score high, positively contribute to the decision-making process. A feature importance score is simply the outcome of applying a specific post-hoc explanation method. These methods can vary widely in the manner in which they arrive at such an evaluation. In some cases, gradient-based information is used; in others, local or global surrogates are employed. Below, we will present representative methods for each approach.

3.1.3.1 Gradient-Based Visualisations

The use of so-called activation-maximisation methods is becoming increasingly popular as a technique for gaining insights into the inner workings of Deep Neural Networks. This process involves synthesising an input that highly stimulates a neuron within the network. By leveraging a deep generative network that is designed to take advantage of prior knowledge, Nguyen et al. (2016) report improvements to conventional activation maximisation methods that solely synthesise neurons based on gradient information. It is claimed that this network is capable of generating truthful synthetic images, revealing learned features in a comprehensible manner, and is generalizable to different network architectures. The interpretability is achieved through the generation of realistic images which maximise feature activation and demonstrate the features learned by a neuron. However, the current application focuses on image data and convolutional neural networks.

In contrast to focusing on features within the model representation, the vast majority of methods attempt to understand a model's decision in terms of its inputs. Here, the variation is to modify the optimisation procedure, such as back-propagation. A popular example is layer-wise relevance propagation (LRP) (Schwarzenberg et al., 2019; Nie et al., 2018).This algorithm is based on propagating a prediction by the neural network backwards into input space, using a set of specially designed propagation rules. Neurons receive a share of the network's output, which is then divided evenly among predecessor neurons until it reaches the input layer (Montavon et al., 2019). The LRP algorithm has been applied to computer vision tasks where it is efficiently displayed as heatmaps and masks illustrating the positive and negative relevance of pixels within an input image.

There are many similar methods that have gained a lot of traction, including back-propagation or gradient-based approaches. Often these methods are optimised for, but theoretically not limited to, visualisation-based systems for image classification. The distinction between the gradient-based methods is primarily determined by how the output score is propagated back through the activation functions, such as the rectified linear unit or sigmoids (Nie et al., 2018). Most of these popular local interpretability methods produce saliency maps which illustrate predictively influential regions in the input image rendering them limited to image recognition tasks.

Gradient-based methods have the advantage that they are easily adaptable to different network architectures. Additionally, they are typically faster to compute than other model-independent methods. Particularly true in the case of the image domain, in which explanations are often visual, and we readily recognise what is relevant to the decision of the model. By highlighting only important pixels, it is simple to identify the relevant parts of an image. A number of other methods, such as LIME and SHAP, can be used to explain image classification, but they are more expensive to compute.

The problem with most interpretation methods is that it is difficult to determine whether an explanation is correct and a great deal of the evaluation is strictly qualitative. A study was conducted to investigate whether saliency methods are insensitive to model and data characteristics (Adebayo et al., 2018). This would indicate that the explanation is not related to the model and data, so insensitivity in this context is not desirable. The article demonstrated that many gradient-based visualisations failed their insensitivity check acting similarly to edge detectors. Edge detectors in this context simply highlight strong pixel colour changes in an image; they are not dependent on a prediction model or abstract features underlying the image, and do not require any training.

3.1.3.2 Statistical Insights Into Features

In this category of methods, the black-box model is queried and statistical metrics are extracted for feature importance or other descriptive statistics in order to provide information about the prediction procedure. Research into efficient statistical analysis has provided numerous methods for gaining insights into a model's behaviour by altering inputs. An approach that is commonly used is the partial dependence plot, also known as PDP. The PDP chart illustrates the marginal effect that one or more features have on the prediction capability of the machine learning model (Friedman, 2001). The partial dependence plot reveals linearity, monotonicity, and more complex dependencies between the feature and the target prediction.

This metric also has a local equivalent, the individual conditional expectation plot (ICE). In contrast to the PDP, which provides a comprehensive overview of the model as a whole, the ICE plot shows the interaction effects between features within a single instance of data (Goldstein et al., 2015). Resulting plots contain one line for each case separately, whereas PDP depicts the average of all these individual lines.

Recently, the Shapley value has gained popularity, originally proposed by Shapley (1953). This approach has been specifically adapted to Machine Learning models and tasks by Štrumbelj and Kononenko (2014) and was popularised by the effective library by (Lundberg and Lee, 2017). We introduced the method in section 2.4.2.The objective is to capture the average marginal contribution of a feature value across a selection of all possible combinations of features. Shapley values for each feature of a specific input indicate its relative contribution to the prediction in relation to the average prediction for this dataset (Lundberg and Lee, 2017). In order to determine this value, the authors propose to establish a baseline by selecting a subset of random feature values from the dataset and predicting using those values. By predicting with this baseline and without the relevant feature, the method produces marginal contributions from the feature. Shapley values are effective for both classification and regression tasks.

Due to SHAP's ability to compute Shapley values, all the advantages associated with Shapley values are available. It is based on a solid theoretical foundation in game theory. Predictions are distributed fairly across the feature values, allowing us to obtain contrastive explanations that compare the prediction to the average model prediction. In addition, a large number of Shapley values can be calculated, enabling global interpretations of the models. There are several global interpretation approaches, including global feature importance, feature dependence, interactions, and summary plots. The global interpretations in SHAP are consistent with the local explanations, since they are built upon them.

However, applying SHAP to neural networks can be computationally costly and time consuming. As a consequence, the approach is not suitable for computation of Shapley values for numerous instances. Therefore, a number of global SHAP methods may not be computationally feasible for these applications, including global SHAP feature importance.

Moreover, feature dependence is ignored by the neural network adaptation of SHAP. Many other permutation-based interpretation techniques suffer from this limitation. Since feature values are typically replaced with values from random instances, it is usually easier to select samples from the marginal distribution. Nevertheless, if features are dependent, such as correlated, then a large amount of weight will be placed on data points that are unlikely.

The counterfactual explanation is another ML-adapted approach that has recently attained increased prominence. The explanation is communicated by referring to necessary changes in inputs to produce an alternative outcome prediction (counterfactual). Following the publication of Wachter et al. (2017), counterfactual explanations gained traction as the authors argued for counterfactual reasoning in the context of the GDPR (European Union, 2016). Most methods involve a loss function that takes a specific instance and a counterfactual class referring to the desired outcome as inputs.

Wachter et al. (2017), for example, presents the discovery of a counterfactual based on a loss that contains two components. One component is the quadratic distance between the model prediction for the counterfactual instance and the desired outcome scenario that must be determined by the user in advance. It is followed by a second term that measures the distance between the observed instance and the counterfactual instance. Consequently, the resulting loss is a measure of how far the predicted counterfactual outcome diverges from the user-desired outcome and how far the counterfactual instance is from the input instance.

This method has the advantage that counterfactual explanations can be interpreted in a straightforward manner. A prediction changes to the predefined prediction if the feature values of an instance are changed based on the counterfactual. It is not necessary to have access to the data or model to use the counterfactual method. Access to the model's prediction function is all that is required.

On the other hand, you will typically find multiple counterfactual explanations for each instance. For some scenarios, this can prove to be problematic as selecting the appropriate explanation can prove challenging.

3.1.3.3 Surrogates

Alternatively, surrogates are external models or explanators that approximate the black-box model. These surrogates are capable of different representations. The purpose of (interpretable) surrogate models is to approximate the predictions of the underlying model as accurately as possible while at the same time being interpretable. It is again possible to differentiate between local and global surrogate models. Global surrogate models are interpretable models that are trained to approximate the predictions of black-box models and are transparently observable in their entirety. Through the interpretation of the fully comprehensible surrogate model, we obtain information about the behaviour of the black box model. Local surrogate models, on the other hand, serve as interpretable models that are intended to explain individual predictions of black box machine learning models.

A popular method in this context is LIME, which stands for Local Interpretable Model-Agnostic Explanation (Ribeiro et al., 2016). As the name implies, these explanations describe specific instances through an approximation of the local factors in the vicinity of the decision boundary of the instance. This approach explains a particular input instance's decision using the weights of a local linear classifier. In order to learn a linear classifier, we need to create a new dataset that contains perturbed samples and the corresponding predictions of the underlying model we are trying to understand. Based on the proximity of the generated data to the instance of interest, the generated data is then weighted during training of the linear classifier. In this approach, the model is trained according to a local fidelity measure, which is attempting to replicate the outputs of the model in a local area around the instance of interest. This approach has the advantage of being agnostic towards the underlying model.

Yang et al. (2018) presents an interpretation tree based on local explanations for several Machine Learning models under the title Global Interpretation via Recursive Partitioning (GIRP). This insight is developed by identifying and extracting a binary tree known as an interpretation tree, which presents a set of decision rules which approximate the original machine learning model. The method partitions the input space in such a way as to maximise the variable contribution from local explanations within each divided area. Given their dependency on contribution matrices, which are extracted using local explanations across a whole dataset, explicit features associated with individual contributions are preferable, i.e., tabular data rather than images and text. Despite its alleged global nature, the method can be seen as an aggregation method of local explanations.

Krishnan and Wu (2017) propose a method called Partition Aware Local Model (PALM). It is intended to help debug Machine Learning models by identifying and summarising the dataset structure. The method employs a meta-model to separate the training set into partitions, and then sub-models to mimic the pattern for each partition. The meta-model proposed by the authors refers to a decision tree that identifies the structure and produces rules that can be revised and aligned with domain experts. The various sub-models associated with the leaves in the tree, on the other hand, capture abstract patterns by using more complex models. The initial proposal is agnostic as to the type of model selected, although the form in which the information is presented for interpretation can differ depending on the type of data involved.

Several adaptations have been made to surrogates to facilitate the adaptation to a variety of tasks. In fact, they can be configured in a variety of ways and can be used to represent different types of information. Applications reach from image recognition tasks (Ribeiro et al., 2016) to responsible gambling (Sarkar et al., 2016). It is important to select models and methods based on specific use cases, applications, and requirements associated with each scenario. A critical factor for surrogates that is frequently overlooked and underreported is how accurate the representations are in relation to the approximated black-box model. They provide the advantage of generality, since they are independent of the black box's underlying model or technique.

Since any model from the interpretable models section can be used as a surrogate model, this approach is highly flexible. Furthermore, it is possible not only to exchange interpretable models, but also underlying black box models. The flexibility of these methods extends to the wide range of data domains that can be explained using surrogates. For instance, LIME is one of the few methods that can handle tabular data, text, and images. In interpreting the black box predictions in the neighborhood of the data instance of interest, the fidelity measure (i.e. how well the interpretable model approximates the black box predictions) gives us a reasonable indication of the model's ability to explain the black box predictions.

Despite this, for local surrogates, defining the neighborhood accurately is a challenging and unsolved problem. Typically, practitioners experiment with different kernel settings in order to find an appropriate neighbourhood setting by observing whether the resulting explanation makes sense. It does, however, exacerbate confirmation bias and cannot be considered as a reliable indicator of appropriate explanations.

In general, one must be aware that conclusions are drawn about the model and not about the data, as the surrogate model does not see the actual label. In order to confidently conclude that the surrogate model is close enough to the black box model, it is not clear what the appropriate cut-off for R-squared or fidelity is. Lastly, any interpretable model you choose as a surrogate will come with both advantages and disadvantages.

3.1.3.4 Distillation for Interpretability

The concept of knowledge distillation refers to the method of transferring hidden knowledge that has been learned by a teacher model to a student model that is typically less complex. As proposed by Hinton et al. (2014), *dark knowledge* refers to salient information that is hidden within logits in the output layer, which is also referred to as the soft targets. Soft targets are the probabilities of all output classes predicted by the model and not just a single output class, which the authors refer to as a hard target. As a means of increasing the information exchange between the models, the authors propose an additional parameter, T, which controls how "soft" or evened out the probability distributions for all classes will be.

Craven (1996) can be regarded as one of the pioneering efforts in converting neural networks into a sparser and more interpretable representation and on distillation for interpretability. It is important to note that while this method uses a more complex query mechanism and uses specific constraints in order to obtain a more accurate imitation of input-output behavior, using the oracle mechanism as a tool in the information transfer step is consistent with all the following distillation methods.

Frosst and Hinton (2017) propose the soft decision tree (henceforth; SDT) as a surrogate because it can be easily adapted to conventional distillation techniques since the model has probabilistic characteristics. Their study illustrates the benefits of using SDTs in visual tasks where logical rules are ineffective. Using this approach, the model learns a hierarchy of filters that are utilized to assign each example to a particular tree branch with a specific path probability, and for each split, it learns a simple, static distribution over the range of output classes, k.

Che et al. (2016) implement Gradient Boosted Trees to distill the knowledge from a Neural Network. The interpretability of the resulting GBT, however, is dependent on a meaningful post-hoc explanation method, namely partial dependence. They also use plain logit models without the mechanism to utilise the hyper-paramter temperature, as well as using soft targets. Tan et al. (2018b) and Tan et al. (2018a) utilise GBTs in addition to Generalised Additive Models (GAMs) using the logits as opposed to the soft targets. In this method, the teacher is a neural network, however their experiments have been limited to binary classification problems.

Dhurandhar et al. (2018) proposes a method for transferring information between a teacher and a student by utilizing confidence profiles, an approach related to distillation. The assumption is that the neural network represents a high-performing teacher, and we can use some of its knowledge of the domain to instruct a simple, interpretable, but generally underperforming student model. Weighting the data points by their classification difficulty can assist the simple model in focusing on easier samples that it can successfully classify during training and, therefore, achieve better performance overall. To achieve this, the authors assign weights to samples depending on how difficult it is for the network to classify them, and do so by introducing probes. Throughout the network, each probe receives its input from a hidden layer and has one fully connected layer with a softmax layer equal to the size of the network output attached.

Xu et al. (2018) integrates distillation and dimensionality reduction into a visual representation of a Neural Network classifier. This method embeds datapoints in a low-dimensional space in order to simplify the transference of the information into a simpler and more interpretable classifier model. After this, the authors graphically represent data points that are assigned similar probability vectors to enable monitoring of the decision-making process, thereby aiding in the diagnosis of deep classifiers.

In general, the benefits and limitations of the surrogate methods also hold for distillation approaches (see end of section 3.1.3.3). In principle, distillation methods can replicate the behaviour of the underlying models more closely. The reason for this is that soft targets provide more information than solely reproducing the output prediction. There is still uncertainty, however, at what point we can consider the explanations to be sufficiently close to the underlying model, and therefore determine the effectiveness of the distilled surrogates.

3.2 Analysis & Taxonomy

Below is an analysis and table that summarises the methods that have been introduced and described above. By providing this overview, practitioners should be able to make an informed decision based on their specific requirements. There are additional dimensions of taxonomy that could have been included, however this overview is centred around the dimensions that are most descriptive in distinguishing the different approaches. We will outline in subsection 3.2.1 that there are potential future dimensions that are not currently being considered but are important for effective holistic XAI approaches from the authors' perspective.

Because the majority of introduced XAI methods do not explicitly state a comparable measure of quality or truthfulness of the explanation, this dimension is not included in the overview table. The majority of methods rely on specific heuristics and measurements to determine the optimal explanation based on the given technique. This dimension would not assist in drawing comparisons in the overview table, therefore it is omitted. However, the quality is referred to in the discussion of individual approaches. Furthermore, the complexity of the algorithm is also rarely reported, and a rigorous comparison of computational cost is outside the scope of this review. We have nevertheless identified several problematic scenarios in the preceding section of this chapter. Following the practice employed in the preceding chapter, we have ordered the list by the stage in which the method is applied. Table 3.1 provides an overview of the categories of the different dimensions, and Table 3.2 applies such categorisation to different practical approaches.

The problem of presenting information in a manner that is human-perceivable has been the subject of extensive research efforts. A visual representation of data structures pre-training include hierarchies, clusters, local sparsities, outliers, and correlations within the data set (see section 3.1.1). Techniques that visualise and summarise data sets enable a rapid and intuitive understanding of their structures and relationships. Such underlying data structures will be leveraged extensively for purposes of inference by the machine learning system. Hence, an understanding of the underlying structure and relationship is critical to determining whether the model's answers will be reliable.

Representative instances and dimensionality reduced representations facilitate the understanding of complex relationships in datasets in an effective and intuitive way to achieve high expressive power. The majority of these approaches are model-agnostic and can be applied to complex datasets allowing for high portability. To take advantage of the specificities in each data domain, however, specific techniques may be required. Many visualisation techniques provide a comprehensive overview of the entire dataset. Further, many tools allow for user interaction by allowing users to explore specific entities in greater detail. Alternatively, methods can also be applied to specific subsets of the data. However, the explanation presented itself does not take the model into account, and therefore should only be viewed as an explanation of the AI system in conjunction with an inherently explainable method or a post-hoc XAI method.

The inherently interpretable methods introduced in section 3.1.2 involve compromises related to linearity, monotonicity, and feature interaction. Through, for example, the use of linear behaviour and the breaking down of complex scenarios into simple rules for handling inputs, SLIM enhances trust in the system. In the same way, rule-based systems facilitate understanding by creating straightforward rules that can be understood easily by users. A relationship of trust is frequently established when

Dimension	Categories
Portability	Agnostic, NNs, RFs, NA (when inherently interpretable)
Locality	Global, Local
Expressive Power	Decision Tree (DT), Decision Rules (DR), Features Importance (FI), Saliency Masks (SM), Sensitivity Analysis (SA), Partial Dependence Plot (PDP), Prototype Based (PB), Sparse Visualisation Based (dimensionality reduction etc.) (VI)
Stage	Pre-Training, Inherently Interpretable, Post-Training
Use-Cases	Various, Tabular, Image, Text
Translucency	Pedagogical, Decompositional, Individual

Table 3.1: Overview of the different dimensions in which we categorised the different methods in Table 3.2.

Method	Portability	Locality	Expressive Power	Stage	Use Cases	Translucency
MMD-critic - Kim et al.	Agnostic	Global	VI	Pre	Various	Individual
t-SNE - van der Maaten and Hinton	Agnostic	Global	VI	Pre	Various	Individual
BRL - Letham et al.	NA	Global	DR	Inherent	Tabular	Individual
PS – Bien et al.	NA	Local	PB	Inherent	Various	Individual
mLSTM – Radford	NA	Local	FI	Inherent	Text	Decompositional (Ind)
GAM – Caruana et al.	NA	Global	SA/PDP	Inherent	Tabular	Individual
SLIM – Ustun and Rudin	NA	Global	SA/PDP	Inherent	Tabular	Individual
Tree-Reg – Wu et al	NN	Global	DT	Inherent	Various/Sequential	Decompositional
TbD-Net – Mascharka et al.	NA	Global	SM	Inherent	Image	Individual
IDS – Lakkaraju et al.	NA	Global	DR	Inherent	Tabular	Individual
Trepan - Craven et al.	NN	Global	DT	Post	Tabular	Pedagogical
CAM - Zhou et al.	NN	Local	SM	Post	Image/Video	Decompositional
LRP - Nie et al.	NN	Local	SM	Post	Image/Text	Decompositional
Act. max. – Nguyen et al.	NN	Local	PB	Post	Various	Decompositional
LIME - Ribeiro et al.	Agnostic	Local	FI	Post	Various	Individual
GIRP - Yang et al.	Agnostic	Global	DT	Post	Various	Individual
PALM - Krishnan and Wu	Agnostic	Local	DT	Post	Various	Pedagogical
SDT - Frosst and Hinton	NN	Global	DT	Post	Various/Images	Pedagogical
Distillation - Xu et al.	NN	Local	SM	Post	Image/Video	Pedagogical
BCM - Kim et al.	Agnostic	Local	SM	Post	Various/Image/Text	Individual
Palm – Krishnan et al.	Agnostic	Local	PB	Post	Various	Pedagogical
1Rule – Malioutov et al.	NA	Global	DR	Post	Tabular	Individual
PDP plot – Friedman	Agnostic	Local	PDP	Post	Various	Individual
ICE plot – Goldstein et al.	Agnostic	Global	PDP	Post	Various	Individual
Shapley – Mascharka et al.	Agnostic	Global	FI	Post	Various	Pedagogical

Table 3.2: Summary of the reviewed methods and their characteristics.

predictive accuracy is reasonable and extracted information matches human domain knowledge and is consistent with expectations.

An inherently interpretable model is one that consists of functions that are monotonic, linear, and does not take into account very complex feature interactions. Nevertheless, there are approaches such as GAMs that can account for nonlinear functions that are complex. Further, rule-based methods can be applied to model nonlinear, nonmonotonic phenomena that are highly multifaceted. However, this may compromise the comprehensibility.

As these methods rely on standalone models, there are no restrictions regarding portability and model specificity. With respect to locality, rule-based methods are, in principle, both globally and locally interpretable. The same applies to regression-based interpretable models, such as SLIM and GAMs.

These methods are situated at different points of the accuracy-interpretability trade-off spectrum. There are complex and accurate models that are difficult to interpret, as well as simple and explainable models that are less effective. There may be compromises made with these explainable models when it comes to task complexity, such as assuming linearity or monotonicity. Although the model provides simplified interpretations for the tasks it is intended to solve and allows for straightforward interpretation, it may be ineffective at solving tasks in more complex scenarios. The GLM, with its different extensions, such as the use of neural networks Agarwal et al. (2021), illustrate that often once models are adapted to more complex tasks, this is usually done at the expense of explainability.

The purpose of post-hoc XAI methods is to facilitate complex modelling, while providing insight that can be easily interpreted. In order to increase the level of trust in Machine Learning systems, metrics such as feature importance need to align with expert domain knowledge as well as meet reasonable expectations regarding the underlying decision-making procedure. In addition, ensuring stability under small data perturbations, as well as demonstrating acceptable predictability under simulated scenarios, is required to create a more credible and trust-worthy outcome.

Similarly, trust is enhanced by surrogate models when interactions, coefficients, and variable importance follow the understanding of domain experts and the explanation remains consistent.

Through the use of gradient-based visualisations, the internal structures of complex models may be exposed, helping to enhance trust in the localised internal mechanisms. Gradient-based visualisations can be used to explore machine learning models of various complexity levels, but these methods likely excel in nonlinear, non-monotonic models. The expressive power here is typically bound to the expressiveness of individual input features, since these are often used as information representation. In order to obtain intuitive explanations, certain methods, however, take advantage of simplifications in the approximation process. This may harm the overall truthfulness of the underlying model.

We have introduced both model-specific and model-agnostic methods. Surrogates are typically agnostic, however, they are sometimes used to exploit the internal structure of methods based on their adaptation to particular models

Several local and global methods have been discussed, depending on the requirements, methods can be chosen accordingly.

3.2.1 Identified Gaps

In our review of explainability approaches, we have seen that the different methods employ a broad range of techniques to provide a better understanding of AI systems. It is apparent from the application domains of some of the reviewed methods that many safety-critical or sensitive applications involve the use of machine learning systems. These applications cover a wide range of areas from healthcare to law and finance. In such sensitive areas, it is imperative to ensure that systems do not unwittingly discriminate against a particular group of people so as to maintain fairness in machine learning models. A significant obstacle is the potential for models to discriminate inadvertently against protected attributes due to hidden correlations. Feature importance scores at the local level may indicate discrimination in certain circumstances, particularly if classifiers are relying on sensitive features, even if these correlations are hidden. However, one fundamental limitation of the current methods is their inability to address any undesirable properties that are exposed by explanation methods. Using a comprehensive neural-symbolic integration approach, which we will outline in chapters 5, 6 and 7, this shortcoming may be addressed by providing ways by which the decision-process can be influenced depending on what information has been extracted. Unlike neural-symbolic approaches to XAI which historically considered the bi-directionality of capturing and inserting knowledge of an underlying system, contemporary approaches almost exclusively focus on capturing knowledge. Even though this approach is beneficial, it may require further development to allow the user to act according to the observations made regarding model behaviour. Upon preliminary review, we conclude that the lack of consideration of directionality may prove increasingly problematic. In order to make best use of XAI methods, practitioners should be able to act on the extracted information through tools that allow for interactive explanation and alteration.

Overall, the majority of methods have been applied to Neural Networks since their superior performance across a variety of domains has made them popular. Though specific methods built from the bottom-up may be more appropriate for specific use cases, the need to better understand the existing models has led to an increase in interest in post-hoc and model-agnostic methods. As a consequence, a promising and effective approach to explainable AI must be compatible with existing systems, so as to keep up with an ever-changing field composed of a wide range of applications.
Nevertheless, this push for general applicability has resulted in many methods lacking the ability to leverage the inductive biases of underlying models and instead being limited to imitating the input-output relationships based on underlying datasets. Despite the fact that flexibility and adaptability are desirable characteristics, we believe that a promising method may require stronger alignment of representation than imitation of input-output mappings or local approximation.

Considering the tradition of symbolic approaches, we believe there are two important pillars of effective communication in explainable AI, namely **representation** and **operation** (Littlejohn, 1977), as will be discussed in the upcoming chapters.

In neural networks, the learned transformation becomes crucial to performing the task. For this reason, a method aiming to represent the model truthfully should also attempt to provide understanding of the model-inherent representation (further discussed in 4.1, 5, and 6.5). Here, we would like to highlight the TCAV approach described in Kim et al. (2018) since it is one of the very few techniques that tries to provide an understanding of intermediate representation based on learned transformation. The lack of consideration of methods that target inner representations may result in a divergence between explanation and model. Based on our preliminary analysis, we conclude that more investigation is needed regarding the appropriate extraction and integration of extracted representations into a comprehensive explanation.

In addition to taking into account the inherent representation, a meaningful explanation should aspire to distill the internal operation in an accurate but understandable manner. We suggest that it may be worth investigating whether it is possible to represent the sequential steps of computation in a neural network in a way that is simpler and more understandable.

Accordingly, identifying the gaps and diverging trends between Neural-Symbolic work and contemporary XAI methods has prompted us to investigate two promising methods that follow the inductive biases of Neural Networks more closely and provide insights using the fundamental building blocks of internal representation and operation. Furthermore, the identified gaps motivated our own approach to XAI that is motivated in chapter 5 and presented in chapter 6 and 7.

Chapter 4

Post-Hoc Explanation for inherent Concepts and Operation

Following an analysis of the existing literature on XAI, we propose to investigate certain XAI methods due to their proximity to a Neural-Symbolic approach. Thus our research will focus on understanding symbolic representation and the underlying operational logic of an AI system. Specifically, we concentrate our efforts on neural networks due to their popularity and inherent complexity. In particular, we aim to determine which limitations prevent these methods from providing an effective approach to XAI. The basic principle of deep neural networks is that they comprise multiple layers arranged sequentially, resulting in the representation of features at multiple levels of abstraction.

As part of our first investigation, we examine the ability to capture abstract concepts that emerge from multiple levels of abstraction within these sequential layers of a neural network. By disentangling the inner representation of the model into concepts of a more abstract nature akin to symbols, as opposed to using the input space as is common practice in XAI methods, a more intuitive understanding may be achieved. Furthermore, focusing on internal representations allows for the extension to multimodal models, increasing adaptability to a wide variety of tasks. It is our objective to explain the different output class mappings that a model has learned based on the importance of the disentangled internal representations. In this manner, we can connect abstract intermediate representations in the form of concept symbols to the output classes of a model. The evaluation of such approach will be centred around the method by Kim et al. (2018).

Following the examination of the inner representation, we intend to investigate the ability to understand model operations. In this context, the term operations refers to the operational logic of the system that specifies how data and representations are interconnected and utilised in order to reach a decision. In order to generate more accurate explanations, we wish to explore a method that utilises an inherent hierarchical structure to mimic the sequential arrangement of layers. Based on the arguments advanced by Nguyen et al. (2021) and Craven (1996), we consider decision trees to be a particularly suitable method for interpreting the sequential logic of a neural network. To accomplish this, we will explore how to extract information from neural networks into tree-structured representations through model distillation. In addition to being inherently hierarchical, tree-structured representations have been demonstrated and used as highly interpretable models that permit auditability. Through translating the model operation into a highly transparent structure, we would be able to understand how features are related to one another for the purpose of accomplishing a task. Through examining features in relation to each other, it is possible to provide more complex explanations, which will be examined across multiple datasets. As part of this analysis, we will evaluate the model distillation into decision trees proposed by Frosst and Hinton (2017). We will focus our efforts on investigating the appropriate means for better understanding the inner operations of the machine. This will be examined by increasing the complexity of the task, and then interpreting the resulting trees to provide information about model behaviour. The examination of representation as well as the investigation of operation will provide a robust foundation on directions towards a more neural-symbolic centric perspective on XAI.

During the evaluation of the existing XAI methods targeting inner representation and operation, the focus should not only be on implementation of the existing method, but also on further development and improvement. This will enable us to judge whether these methods constitute the building blocks for an effective neural-symbolic approach to XAI. In demonstrating the limitations, we go on to argue for a more bottom-up approach to address the XAI challenges identified through the experiments. Throughout this chapter, we will illustrate that XAI approaches that concentrate on either representation or operation are fundamentally limited. Only when both are considered concurrently can an effective basis for a Neural-Symbolic approach to XAI be established.

4.1 Explaining Neural Networks using Concept Activation Vectors for Concept groundings

The lack of methods that target the inner representation is one of the gaps identified in the section on the taxonomy of explainability. A key success factor of Deep

Learning algorithms is their ability to convert low-level data into representations that enable models to perform task solving. Therefore, it is important to determine if it is possible to extract such representations in a meaningful way. However, as the representations are highly distributed and learned, extracting and interpreting them is not a straightforward task. As far as the author is aware, Kim et al. (2018) is one of the first papers to propose the extraction of abstract concepts in the context of Deep Learning intermediate representation for explaining Convolutional Neural Networks. The focus of their research is primarily on computer vision as a domain of application due to the intuitive nature of the representation. Images depict objects that are formed through the arrangement of specific pixel values. These objects may consist of concepts such as, for example, specific patterns or shapes.

To investigate the appropriateness of this method for reasoning about what has been learned, we will implement this approach into the PyTorch framework in the following experiments and run experiments. The accompanying code is located in the nlp_cav folder of the repository. In particular, we are interested in seeing if the grounding of abstract concept representations can provide sufficient explanations to reason about what the model has learned. Moreover, we chose natural language as a complex data domain for our experiment in order to test the method's adaptability. To do so, we present a novel ontology querying pipeline that automatically learns concept explanations by enriching the concept representation.

Despite extensive research conducted in the area of Machine Learning explainability, there is often no direct connection to the application of Natural Language Processing (NLP). Thus, the majority of NLP explainability methods utilised today are drawn from other data domains and do not take advantage of the advantages afforded by customized approaches.

Currently, among the most popular methods in the language domain are saliency based explanations, which result in feature attribution in the input space (Arras et al., 2016; Schwarzenberg et al., 2019). Aside from their limited suitability for NLP, they are also mostly limited to local explanations. This means that they are designed to explain specific decisions in the context of specific input instances. Moreover, since the input dimension in NLP are often extremely high-dimensional, if words are converted into vectors, feature attribution methods such as those described above would result in too granular and complex explanations. To ensure the effectiveness and validity of the model as well as prevent potential undesired biases, more comprehensive approaches must be developed that provide a means to assess what factors drive the decisions of a system intuitively.

The interpretation of machine learning models poses several challenges due to the fact that they operate on low-level input features that are transformed through non-linear internal transformation using numerous sequential layers. Considering the high performance of neural networks and recent research, these powerful distributed neural networks may be able to capture abstract concepts via non-linear activations which may result in abstractions akin to human understandable concepts, as demonstrated by Zeiler and Fergus (2014). As a result of this, it would be beneficial to capture aspects of higher-level concepts and operations, such as logical rules, in order to provide meaningful explanations. This section will focus on capturing abstract concepts. By applying the proposed method, we will be able to test the output classes of a model based on user-defined abstract concepts for language models applied to the classification problems. This model will produce scores that will allow us to determine what concepts have contributed to the model's prediction of a specific class and by how much compared to other concepts.

However, in the context of NLP, the definition of a concept is not easily discernible. It is important to note that isolated words are already concepts that are refined when observed in context as part of sentences. Nonetheless, our approach goes beyond simple word importance. We will examine word groups associated with a concept and used in multiple contexts with the same meaning. One example would be the concept of *family* that incorporates words like *mother*, *father*, *daughter*, *son*, *relatives*, and more. Each output class will be assessed on the importance of talking about family. Another way in which language is used abstractly in practice is by using specific forms of expression to convey abstract ideas. A good example of this is sarcasm, where the true meaning of the words are not intended to be taken literally. In order for a model to account for abstract patterns such as these, a latent representation must be utilised.

4.1.1 Related Work

The approach we use in this section closely follows the methodology outlined in Kim et al. (2018). In this paper, the authors introduce the notion of Concept Activation Vectors (CAV). Essentially, a CAV represents a vector pointing to the direction of activation values associated with representative datapoints of the concept in a hidden layer. Through the use of this vector, we can compute the sensitivities of output classes in relation to specific concepts. For example, given a model that identifies zebras and an example set that exemplifies the concept "stripe patterns", TCAV (Testing with Concept Activation Vectors) is able to quantify the impact of stripes on the prediction of a zebra as a single number. Here, TCAV refers to the general method, and CAV refers to the explicit vector representation that is used to extract the concept.

Recently, it has been proposed to use convolutional kernels for similar purposes (Townsend et al., 2021). In this paper, the authors arrive at an approximate description of a CNN's behaviour by employing logic rules that connect to convolutional kernels. The authors suggest that, as a result of this connection to convolutional kernels, the semantic concepts they represent are incorporated. It is shown that the kernels identified within the last convolutional layer may be referred to as symbolic in that they exhibit strong reactions to similar input images that effectively divide output classes into sub-classes.

Therefore, both approaches emphasise the use of latent model representation in order to extract abstract concepts. We will investigate the potential for an adaptation of the TCAV approach in the linguistic domain as the authors confine their study to the image domain. As a result of the intuitiveness of high-level concepts within an image and their relationship to pixel values, an approach that would generate explanations based on concepts and not numeric pixel values (using saliency maps) would be desireable.

Translation and adaptation of the approach for the NLP domain, however, may provide a number of advantages over other the image domain. The semantic interpretation of the embedding spaces in NLP has been studied for some time. In fact, Kim et al. (2018) relate their idea to NLP techniques such as Word2Vec in order to motivate the use of semantically meaningful directional vectors in the embedding space (Mikolov et al., 2013). The depiction of the word embeddings and mappings in Figure 4.1 illustrates the ability of vector embeddings to capture abstract concepts in embedding space by means of distributed learned representations. Therefore, this is intuitively a compelling reason to use the TCAV method, which is based on the extraction of semantically rich embeddings, in the domain of natural language.

Our adaptation of the TCAV method also incorporates the integration of ontologies in order to arrive at concept groundings. As will be outlined, this enables the extraction of concepts from the model representation more effectively by using commonsense knowledge.

4.1.2 Method & Experimental Setup

As already stated, the basis of the approach is TCAV by Kim et al. (2018). The method here is adapted to be used in the language domain and we developed a

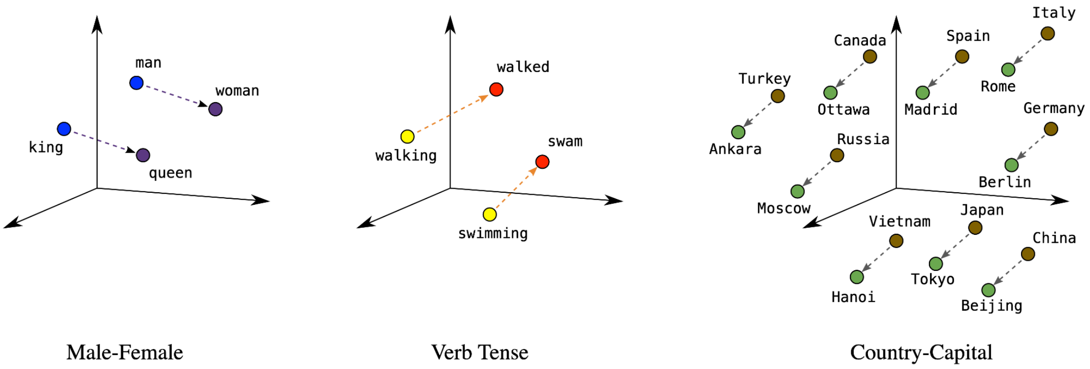


Figure 4.1: Illustration of the semantic meaning of rich vector embedding spaces (GoogleDevelopers, 2021). Here it is demonstrated how the embeddings may be used to produce powerful analogies.

customised pipeline that allows for ontology integration. In this section, we will summarise the method and explain our adaptations and relate to the original TCAV method.

We consider every Neural Network with inputs $\boldsymbol{x} \in \mathbb{R}^n$ and any layer l within the network consisting of m neurons, as proposed by Kim et al. (2018). As a consequence, data inference as projection into layer l can be expressed as follows:

$$f_l : \mathbb{R}^n \to \mathbb{R}^m$$

In order to gain a more comprehensive understanding of how the network functions, we will make use of the latent representations inside the network. A layer l that will be used to extract the concept representation must be selected in advance.
While the choice of layer for extraction is fully customisable, in NLP it is advisable and sufficient to choose an early layer when using commonly used pre-trained word vectors. Due to the fact that these embeddings have already captured semantic information, it is not necessary to select particularly deep layers, as is the case in computer vision. This differs from the approach taken in Kim et al. (2018) where the results show that the layer l can be detrimental to the identification of meaningful concepts associated with the explanations. Further discussion of this subject will be presented in chapter 7.

4.1.2.1 Defining the Concepts and CAVs

For the purpose of probing the model for user-defined concepts, Kim et al. (2018) suggests collecting a set of examples representative of such concepts. This can either be done manually or by selecting a dataset that includes labelled concepts. Note that there are no prerequisites for the training data of the concept as well as the model architecture. As a result, we can test the trained model on concepts that were not included in the initial training set in order to obtain insight into the generalisability of the model. The capability of the method to extract meaningful concepts is dependent on the successful optimisation of the network based on initial conditions, meaning that we assume that the network has learned concepts using common ML practice (such as cross-validation and regularisation).

For our NLP adaptation of this approach, we employ a pipeline that is capable of extracting conceptually related words and automatically populating sets of examples based on these words. It is accomplished by querying an external Knowledge-Base to find words that are relevant to the specific concept. We utilised ConceptNet which is a semantic network composed of over over 1.6 million assertions of commonsense knowledge encompassing the spatial, temporal, social, physical, and psychological aspects of everyday life (Speer et al., 2017). We query the ConceptNet ontology to capture a rich set of appropriate related words solely based on a single concept keyword. Figure 4.2 illustrates this for the concept of family.

When the appropriate words have been extracted, we perform an automatic search of the entire dataset to identify sentences that contain any of the concept-related words. In the current implementation, the n-gram size surrounding the word can be determined by the user. Furthermore, we need to populate a negative set, ideally of equal size, derived from sentences that do not include any words associated with the concept of interest.

We may also use the same data aggregation process as Kim et al. (2018) and utilise other datasets for the testing of concepts. Probing the network about concepts that are not part of the training set is an important feature of this approach. An important component of this method is identifying the Concept Activation Vector. Specifically, this vector corresponds to a vector within the activation space of the user-defined layer l, which represents the concept of interest. With the positive and negative sets of the concept obtained, the CAV is then calculated by training a linearly separating hyperplane that distinguishes between examples containing and excluding the concept. Once a hyperplane is found, the CAV is merely the normal of the hyperplane. One can achieve this by training a binary linear classifier on the activations of the sets in the hidden layer of the network. This process is depicted in Figure 4.3 and will be illustrated using an example below.

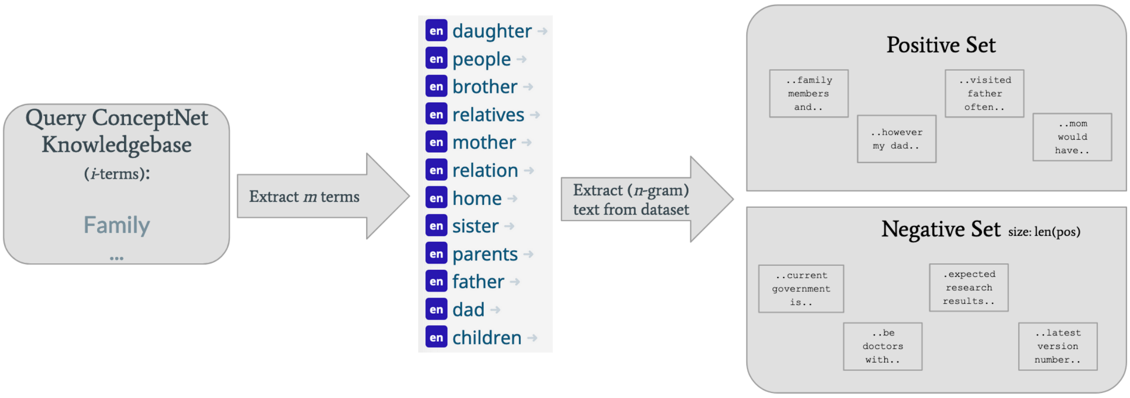


Figure 4.2: Illustration of the ConceptNet ontology integration to automatically create sets for the extraction of concept activation vectors. Using the *RelatedTo* relation, we can derive words from the same concept and create more comprehensive sets than following the approach proposed by Kim et al. (2018).

4.1.2.2 Sensitivities and Derivatives

In many of the interpretability methods outlined in the taxonomy, explanations are derived through computing the gradient of the output logit values with respect to the input features. As a consequence, the explanation will always reside within the input space, such as the importance of individual pixels or word embedding dimensions.

$$\frac{\partial h_k(\boldsymbol{x})}{\partial \boldsymbol{x}_a}$$

where $h_k(\boldsymbol{x})$ is the logit for a document x and x_a is a single word in document x. These saliency methods take advantage of such derivatives to assess the sensitivity of different output classes to changes in the embedding of word a.

However, in this method we attempt to measure the sensitivity of inputs to the direction of a concept that exists at hidden layer l. We define $\boldsymbol{v}_C^l \in \mathbb{R}^m$ as the unit CAV for the concept C in hidden layer l. The activations for each input $\boldsymbol{x}$ at hidden layer l are denoted as $f_l(\boldsymbol{x})$. Following (Kim et al., 2018), we calculate the sensitity of concept C with respect to class k as the directional derivative $S_{C,k,l}(\boldsymbol{x})$ where $h_{l,k} : \mathbb{R}^m \to \mathbb{R}$ maps the activation vector to the logit output of class k:

$$S_{C,k,l}(\boldsymbol{x}) = \lim_{\epsilon \to 0} \frac{h_{l,k}\left(f_l(\boldsymbol{x}) + \epsilon \boldsymbol{v}_C^l\right) - h_{l,k}\left(f_l(\boldsymbol{x})\right)}{\epsilon} = \nabla h_{l,k}\left(f_l(\boldsymbol{x})\right) \cdot \boldsymbol{v}_C^l$$

As such, we are not computing this metric on a per-word basis, but on a per-concept basis on a set of words. Once again, we adhere to Kim et al. (2018) by computing the directional derivatives for different classes of input. This means that we calculate the sensitivities of the model we are attempting to explain for each output class. The TCAV score is defined as follows:

$$\mathrm{TCAV}_{Q_{C,k,l}} = \frac{|\{\boldsymbol{x} \in X_k : S_{C,k,l}(\boldsymbol{x}) > 0\}|}{|X_k|}$$

where k denotes one of the class labels in our dataset, and X_k represents all inputs associated with that label. The score is then defined as the fraction of inputs from class k whose activation in layer l was positively influenced by the concept C under consideration, resulting in a $\mathrm{TCAV}_{Q_{C,k,l}} \in [0, 1]$.
We follow the implementation of Kim et al. (2018) where $\mathrm{TCAV}_{Q_{C,k,l}}$ only depends on the sign of $S_{C,k,l}$. Alternatively, the magnitude of the conceptual sensitivities may be considered. The advantage of this approach is the ability to interpret TCAV_Q globally across all inputs for each label.

We also introduce an assessment of the accuracy of separation within the hidden layer l. Here, we only consider CAVs resulting from a linear hyperplane that has high accuracy above the threshold t and can therefore distinguish positive and negative sets according to their activation values. The hyperplane may fail to distinguish between the positive and negative sets of our concepts, which could render the resulting CAV meaningless. Hence, the pipeline will not calculate any scores for concepts that the model may not have learned to represent based on a set accuracy threshold $t \leq 0.8$.

4.1.3 Experiments & Results

Here, we will demonstrate the applicability of our method through the implementation of the pipeline, and thus, introduce the TCAV method to the NLP domain. Our dataset for this application is the widely known newsgroup dataset. Despite focusing on images for the majority of this thesis, because data in this domain lends itself well to intuitive concepts and levels of abstraction, we wanted to contribute something original in this section. As a consequence, we wanted to illustrate how concepts can be applied to the language domain given that the original TCAV paper focuses exclusively on images. Nevertheless, in chapter 7 we also demonstrate some of the shortcomings of TCAV on image data and draw direct comparisons. The reader should refer to Kim et al. (2018) for a detailed examination of TCAV on images.
The NLP newsgroup dataset consists of approximately 18000 posts on 20 topics, divided into two subsets: one for training (or development) and another for testing (or evaluation). Each of the 20 separate newsgroups is associated with a specific topic. Some newsgroups have strong connections (e.g. comp.sys.ibm.pc.hardware /

comp.sys.mac.hardware), while others are highly dissimilar (e.g talk.politics.mideast / misc.forsale).

We train an LSTM model with batch-size of 128, 5 epochs, a sequence length of 20, learning rate of 0.01, hidden layer size of 256 and word embedding size of 300. The code for all the experiments within this thesis is published in the accompanying repository. We use pretrained word vectors for this experiment. The embeddings used in this approach are the so-called ConceptNet Numberbatch embeddings, but any other vector embedding technique could have been used instead (Speer et al., 2017).

ConceptNet Numberbatch is comprised of word vector embeddings that can be used to impose some semantics of word as a starting point for further machine learning efforts. The ConceptNet Numberbatch system is built upon the open data project's knowledge graph we also use for populating the concepts sets, from which the word embeddings are derived. As such, the embeddings benefit from the fact that they have semi-structured, common sense knowledge from ConceptNet. As a result, a method is presented for learning about words that is not based solely on viewing them in the dataset context and determining their meaning from that context. The semantics, which will enable us to identify concepts in the embedding space, will rely on broader knowledge and data.

When training the LSTM model configuration using the Adam optimiser, we obtain 98% accuracy on the dataset. Our subsequent objective is to explain specific output categories using abstract grounded concepts. According to the TCAV procedure, a positive and negative set is generated by utilizing the ConceptNet API. The end-user only has to enter the concept of interest, and the pipeline will populate a set of m terms from the Knowledge Graph using the "RelatedTo" attribute. A schematic representation of the step can be found in Figure 4.2.

With the acquired m terms, we extract text passages in which these terms are used. Although the text passages may be extracted from another dataset, here we rely on the same newsgroup data. After extracting the passages, we populate a negative set, which does not contain any of the initially extracted terms.

This process is repeated for three exemplary concepts: family, hierarchy, and drugs. Our next step is to use the extracted concept sets to derive the concept activation vectors required to calculate the concept importance scores (TCAV). As illustrated in Figure 4.3, we feed the positive and negative sets into the Neural Network. At layer l, also known as the bottleneck layer, we train a linear classifier using the activations of both text input sets. By using the resulting hyperplane, we can extract the orthogonal concept activation vectors for each concept: family, hierarchy, and drugs.

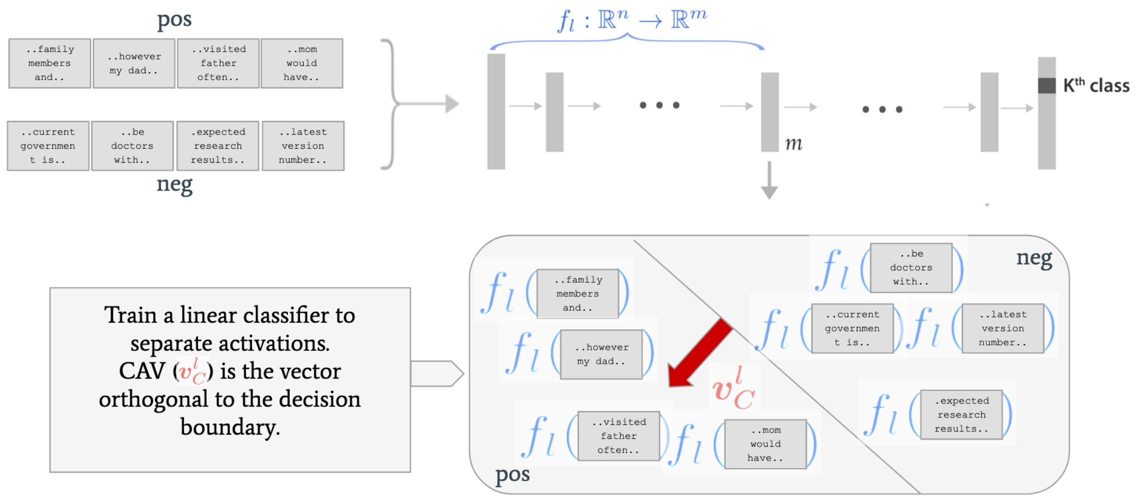

pos
..family members and..
..however my dad..
..visited father often..
..mom would have..
..current governmen t is..
..be doctors with..
.expected research results..
..latest version number..
neg
$f_l : \mathbb{R}^n \to \mathbb{R}^m$
m
K^th class
Train a linear classifier to separate activations. CAV (v_C^l) is the vector orthogonal to the decision boundary.
f_l
v_C^l
neg
pos

Figure 4.3: Illustration of the pipeline that we employ to derive the Concept Activation Vectors. We calculate the m-dimensional activation in layer l of interest using the extracted positive and negative sets. Using a linear classifier, we derive the Concept Activation Vector (CAV) as proposed by Kim et al. (2018).

Subsequently, $S_{family,recautos,l}(\boldsymbol{x})$ provides a quantitative measure of the sensitivity of model output class $rec.auto$ in relation to concept $family$ within a model layer l indicating a pre-concept scalar quantity of an input. This process is then repeated for a set of inputs for each class k (here $k = rec.auto$) and concept C (here $C = family$) in order to calculate TCAV scores. They represent the fraction of inputs belonging to class $rec.auto$ whose layer-l activation vector has been positively influenced by concept $C = family$.

We can use this process to compare the different TCAV concept scores for a specific

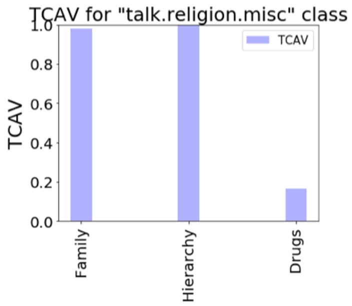


Figure 4.4: Exemplary concept sensitivities for the religion newsgroup class. The figure illustrates the importance of the concepts that the user assigned to the pipeline.

class, as shown in Figure 4.4. This way, one can determine which class outputs are significantly influenced by the various concepts in comparison, thus providing greater understanding of the model.

Figure 4.5 and Figure 4.6 are graphical representations of the concept importance for all 20 output classes. While some results appear intuitive, such as the insignificant contribution of the concept "family" to the class "comp.sys.mac.hardware", others appear to be less intuitive. In this instance, we would be able to investigate by extracting some of the inputs of the class in question whose activation in layer l is associated with C.

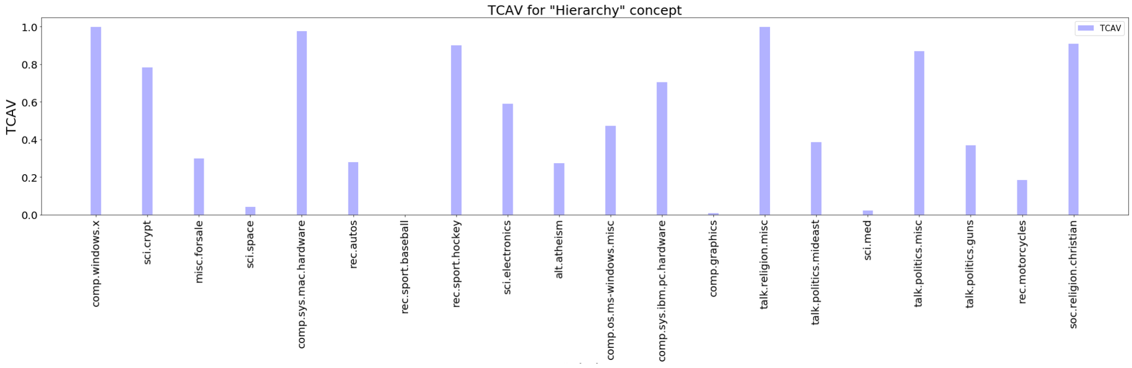


Figure 4.5: As we are inferring 20 different newsgroup classes, an informative information presentation presents the TCAV scores for a specific concept across different classes. Here we check the "hierarchy" concept for all of the different classes.

We demonstrate this with respect to the class *rec.auto* that has received a high score from TCAV for the concept *family*. The following four text extracts provide evidence of why the model emphasised the concept of family. *"fun-to-drive family vehicle", "sedan is designed for mothers", "number of different drivers in a family"*, and *"I had a dream that my dad bought a Viper"*. Based upon the extracted text, it appears that the model has discovered a strong connection of the output class *rec.auto* in relation to the importance of discussing a car's usefulness to the family as context while also highlighting that cars can be a personal matter.

Next, we examine whether these values can be extracted from a more complex linguistic concept. Using the same pipeline as Figure 4.3, we replace the custom ConceptNet sentence extraction mechanism with the *SARC* dataset containing examples of sarcasm (Self-Annotated Reddit Corpus by Khodak et al. (2019)). We attempt to determine whether we can apply the same approach to find out if sarcasm is particularly prominent in specific categories of newsgroup. However, upon training the linear classifier in layer l in order to construct the linear hyperplane, the linear classifier does

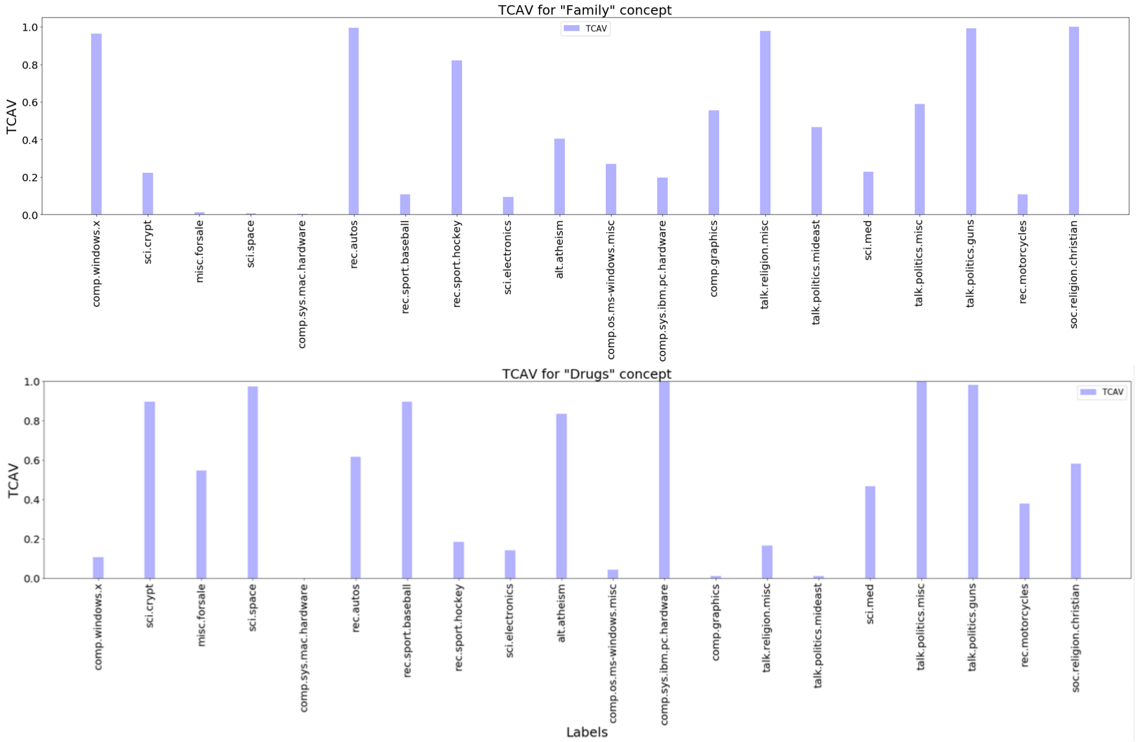


Figure 4.6: Here we show the TCAV scores for the ”family” and ”drugs” concept across all of the 20 different newsgroup classes.

not meet the accuracy threshold $t > 0.8$ that we imposed on the TCAV method. As a result, we are not able to differentiate between sarcastic and non-sarcastic language in the high-dimensional space due to the fact that the model did not learn such complex representation for a relatively simple task of distinguishing the newsgroups.

4.1.4 Limitations of the TCAV approach

In our initial experiment, we were successful in linking the output of the model with high-level concepts. The end-user may be able to consider the appropriateness of the model when connecting concepts to various output categories. This approach has provided sensible results for the proposed model and data and pointed to initially less intuitive connections, such as the connection between the concept of family and the topic of automobiles. By means of abstraction, information can be communicated more effectively, in a manner akin to the understanding of human explanations. Through the enrichment of vector representations through ConceptNet, concept sets can be automatically generated, making the method more accessible.

Nevertheless, there are some major limitations. There is a possibility that the interpretation of the TCAV score may not be as accessible as intended by the original proposers Kim et al. (2017). While the normalised scores provide some insight into the importance of concepts, they are not designed to conform to any specific mathematical properties. In this case, the interpretation is dependent on the TCAV scores of other concepts or output classes.

Additionally, as illustrated by the examples in Kim et al. (2018), the choice of layer l may be essential in the extraction of concept activation vectors. According to the study, depending on which layer the concept is extracted from, there may be a very weak signal, making it difficult to derive meaningful activation vectors. Our introduced threshold t becomes essential in this context as it prevents us from extracting meaningless scores.

Moreover, the layer-dependence may not pose such a significant challenge with our NLP extension since we recommend the use of pre-trained word vectors, which is a common practice today (Qi et al., 2018). Consequently, the word sequences, after having been converted to vectors, are already semantically rich when they arrive at the input layer. Training from scratch as well as learning complex transformations (e.g. in the image domain) with multiple levels of abstraction may make it more difficult to select the appropriate extraction layer.

More importantly, however, the importance of concepts is only assessed individually, and the user is not given the opportunity to connect concepts in order to understand them. An example that would require the combination of concepts may be the discussion of a specific topic with a sarcastic undertone, which would result in much stronger explanations. Isolating the significance of particular abstractions and leaving out the internal operations of the model connecting the abstractions, leaves out an essential component. To provide the user with an effective method to reason about what has been learned, the consideration of operations on top of the learned representations is vital. In the absence of combining the concept representations, the expressive power of the method will remain limited to individual concept sensitivity levels.

Hence, our focus will now shift to a method that concentrates specifically on the operation by distilling the model into a sequential structure. Specifically, we are attempting to discover how features are combined to arrive at a decision, so that we may better understand the internal reasoning process of the model.

4.2 Soft Decision Trees

Knowledge distillation and extraction techniques can be used to reduce complex models into simpler representations in order to achieve high predictive performance in an understandable manner. One of the limitations of classical knowledge extraction methods is that they fail to effectively operate in domains with low-level sensory data. Therefore, current image classifiers and language models are often opaque when it comes to making predictions. Frosst and Hinton (2017) propose to produce visual explanations through the creation of a surrogate model that provides insight into the decisions that are being made. In order to accomplish this, they use a tree-structured hierarchy of representational filters. Tree-based approaches have been purposefully selected for this task as the decision tree classifier is one of the most intuitive approaches to classification in use. In this approach, hierarchical decisions are made in sequence utilising the input features and learned filters. By retracing a decision, a user can quickly understand the reasoning behind the decision, and this serves as a useful conduit between data and human understanding. This facilitates the explanation of not only why a particular decision was made, but also illustrates the broader functionality of the model.

The contribution of Frosst and Hinton (2017) is primarily theoretical as the analysis of the distillation process using Soft Decision Trees is limited in scope and based on a single extracted tree. Consequently, it would be of interest to examine how representative the initial results are and to challenge the approach through various experiments with varying data-domains. Thus, we can determine whether the method is effective in providing a means to reason about what has been learned.

4.2.1 Related Work

Various knowledge extraction techniques may be categorised according to their context in relation to the two main AI paradigms. Firstly, the symbolic perspective strives to establish logical rules that are comprehensible and are consistent with the model. Secondly, the more recent advances in knowledge extraction, which have been derived from the connectionist perspective and attempt to distill a large model into a smaller one.

TREPAN was the first prominent method to use a trained neural network to extract information that could be represented as a surrogate comprehensibly (Craven, 1996). The learning of the surrogate tree itself occurs in a similar manner to the C4.5 algorithm, where the objective is to extract symbolic rules that describe the behaviour of a neural network. More specifically, we build decision trees using so-called *M-of-N* rules, which means that at least M out of N rules must be satisfied for a condition to be met. A greedy search algorithm, which maximises information gain, can be used to generate tree-based structures of such rules. The features can be incrementally added up to the point where information gain is no longer being improved or the size reaches a predetermined threshold. TREPAN combines artificial examples in addition to the dataset to query the neural network and provide sufficient samples for reliable split generation on lower branches.

TREPAN has been extended to incorporate semantic information derived from an ontology in a method proposed as TREPAN Reloaded Confalonieri et al. (2021). The authors include ontologies, which model domain knowledge, in the process of extracting explanations in order to enhance their understanding. They also show that using this semantic information increases human understanding of the model.

While the process of distillation is in many ways similar to the process of knowledge extraction, there are some differences discussed below. To begin with, it is critical to recognise that the connectionist approach originally aimed to develop distillation based on the fact that many of the state-of-the-art neural networks were of considerable size. As a consequence of the large size, computational costs are high. Moreover, in order to deploy models on smaller devices, such as smartphones, reasonably efficient models must be used to compute on smaller devices (Tan et al., 2018a). Due to this reason, distillation is sometimes referred to as model compression.

Distillation's core concept is to train a large accurate model that runs slowly with high computational demands, then distill its knowledge into smaller, faster yet nonetheless accurate models. The process can be characterised as mimic learning since complex models, such as a deep model or a network ensemble, serve as the base model or teacher model to train a student or mimic model. The distillation or mimicking process is characterised by the use of soft labels that are generated by the teacher as target labels for training the student model. These targets fall within the range of $[0, 1]$ and are the output of the teacher model. The main objective of this approach is to overcome the loss of accuracy associated with shallower or smaller models.

It is this sparsity in the smaller student model that is useful for more comprehensively presenting the inner operations of the model. Thus, a variety of interpretable surrogate models have been proposed as student models, such as General Additive Models or Soft Decision Trees (Tan et al., 2018a; Frosst and Hinton, 2017).

Yıldız et al. (2012) proposed Soft Decision Trees as an alternative to common Decision Tree classification. Instead of hard decision nodes deciding which instance goes to each child node based on the outcome of a logical test, a soft decision node routes instances to all its children based on probabilities determined by a sigmoidal function. The resulting decision boundary allows for a smooth boundary in the feature space. Frosst and Hinton (2017) present an implementation of Soft Decision Trees for image recognition. Based on the input-output function of a neuron, the authors use the term $(wx + b)$ as split condition, which represents the input potential of a neuron within a neural network and is also referred to as node mask in this context. In terms of optimisation, the main benefit of such a design is that it is continuously differentiable with respect to all model parameters.

4.2.2 Method

In the Soft Decision Tree, as it is continuously differentiable, we can optimise the parameters using gradient descent on the loss function corresponding to the parameters. To use this approach, it is necessary to forego the hard binary splits that are generally imposed on data when using regular decision trees. After training, each inner node i has a learned filter w_i and a bias b_i. If we consider one example inner node, then we calculate the probability of selecting the branch on the right as follows:

$$p_i(x) = \sigma(xw_i + b_i)$$

where w_i are the weights or filters of the model, σ is the logistic sigmoid function, x represents the input data, and b specifies the bias of the inner node, also illustrated in Figure 4.7. In this manner, each inner node computes the correlation between node masks and the input image, and assigns probabilities for negative correlations on the left and positive correlations on the right.

The model introduced above is related to a hierarchical mixture of experts proposed by Jordan and Jacobs (1994). Frosst and Hinton (2017) refer to each expert as a bigot, since these experts always yield the same distribution after training. As a result, they will always produce the same distribution. Thus, the model will learn a hierarchy of different filters based on which each example is assigned to a particular bigot or leaf with a specific path probability. The static distribution of all possible output classes k computed at the leaf is given by:

$$Q^l_k = \frac{exp(\phi^l_k)}{\sum_{k'} exp(\phi^l_k)}$$

where Q indicates the probability distribution at leaf l. The learned parameters at each leaf are represented by ϕ. The leaf node will provide the *softmax* calculation for

k individual classes. In addition, to prevent overly even (soft) distributions, Frosst and Hinton (2017) apply the inverse temperature β to the activations of each mask node prior to computing the sigmoid function. As a result, the probability of nodes takes the following form:

$$p_i(x) = \sigma(\beta(xw_i + b_i))$$

The probability distributions for the leaf nodes for different paths are computed at the final layer node, as depicted in Figure 4.7.

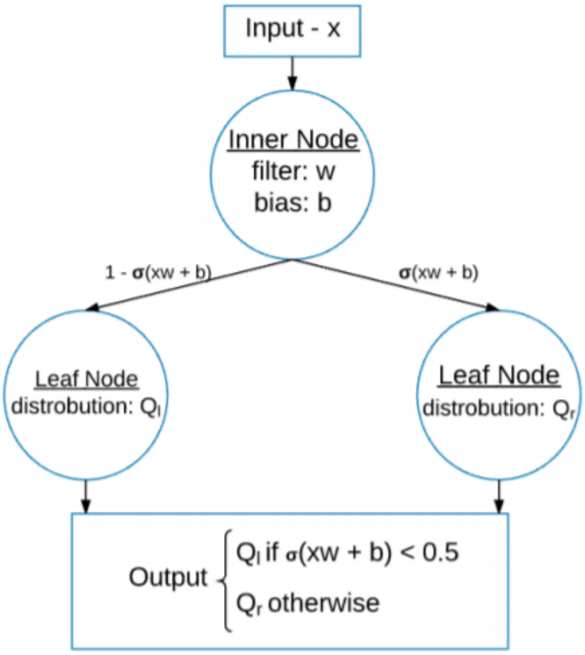


Figure 4.7: Illustration of a Soft Decision Tree with one inner node, two leaf nodes, and the connecting soft splits that feed into the output probability distributions Q_l and Q_r for a binary classification (Frosst and Hinton, 2017).

We minimise the following loss function to optimise the trainable model parameters:

$$L(x) = -log(\sum_{l \in leafnodes} P^l(x) \sum_k T_k log Q_k^l)$$

for a single training example with input vector x and the target distribution T, which can be either a one-hot vector when using dataset labels or soft targets when distilling the soft targets from a teacher model. $P^l(x)$ denotes the path probability at that leaf and Q is the probability distribution for k classes at leaf l. The function is equivalent to the weighting of the categorical cross-entropy.

Consistent use of the input data is an important aspect of ensuring the intelligibility of a model. In spite of the fact that each node of the tree is characterised by a different set of weights and biases, the input vector x remains the same. By comparison, the structure of a general neural network is a sequential progression from the input layer through all layers. Consequently, the output of the previous layer serves as the input for the next layer. However, the input of each split node in soft decision trees is independent of the output of prior layers, enabling a comprehensive understanding of the operation at every level.

It was originally proposed that Soft Decision Trees should make predictions comparable to ensembles by considering the probability distribution of each leaf node. The disadvantage of this approach is that it is impossible to directly trace the decision-making process within the tree as the decision is distributed across the entire tree. Thus, in our adapted approach we opted to make predictions based only on nodes in the tree with the highest probability. It is important to note that this decision is at the expense of a reduced performance in the model.

An initial investigation will be conducted to replicate the Soft Decision Tree on the MNIST dataset, as described in the initial paper (Frosst and Hinton, 2017). Upon replication to a satisfactory level, additional experiments with different data may be considered. Using this new approach, the experiments were conducted with a particular focus on how knowledge is represented and whether it presents a feasible communication method to learn about the model operations. We implement the proposed method of SDT distillation process within Python using the Pytorch framework. The accompanying code is located in the SDT folder of the repository. Experiments are conducted on different datasets to determine the adaptability and to provide more detailed results to the method in general. Beginning with the initial method and experiment put forth by Frosst and Hinton (2017), we examine comprehensibility and performance of the image recognition task as a function of image structure and complexity of the depicted objects. As part of our research, we also investigated the application of the SDT method to alternative data domains, such as tabular and natural language data. However, we believe that the model is fundamentally limited compared to alternative interpretable models that can be applied to these data domains. It is one of the advantages of the node mask to provide information in a visual format for a large number of input dimensions. For tabular data, regular decision trees (or the extraction of them by means of TREPAN) provide a much better level of comprehensibility. For natural language, using word vectors as inputs to the tree would necessitate aggregation per word vector to determine the importance of each word. It is possible to achieve this level of explanation by using inherent self-attention, without resorting to a surrogate, which requires additional approximation. Therefore, our focus in this section remains the intended data domain.

4.2.3 Experiments & Results

If not specifically stated, each experiment was conducted with the given train and test sets split from each dataset source. Further, we obtain a validation set using 10% of training data entries. The best performing model in terms of accuracy on the validation set was selected to calculate and report test set results.

For investigating the effectiveness of the knowledge representation, we perform a number of modifications to the implementation. To ensure that our implementation is compatible with various experiment runs, we track important metrics for comparison and adapt the implementation for wider compatibility to a range of datasets. Furthermore, additional functions were implemented to provide a visual representation of the tree in a way that promotes understanding in addition to saving the decision traces through the three, thus allowing the classification process to be tracked.

4.2.3.1 MNIST

In general, our results are consistent with those reported in Frosst and Hinton (2017). Frosst and Hinton (2017) achieve 94% on the identical test set using a tree built with 8 layers that was trained directly from the data, compared to the 92% using our implementation. Due to the lack of information regarding the implementation of the original paper, it is difficult to determine the reasons for the small difference in accuracy. Additionally, all hyperparameters except tree depth had to be estimated since they had not been published.

The training of 50 epochs of a 4-layer Soft Decision Tree takes 23 minutes using a NVIDIA GeForce 1080, a 8-layer SDT requires more than 160 minutes, and a 10-layer SDT may take up to 10 hours[1]. There is an exponential growth in computation time for larger trees due to the exponential growth in the number of nodes for each additional layer. Hyperparameters are determined using the best performing model on the validation set.

Figure 4.8 illustrates the Soft Decision Tree trained on the MNIST data. The weights are rescaled for the visualisation to the full range (0,255) in order to convey as much

[1] We observe no variation in performance after 45 epochs and made a choice to use 50 epochs for training.

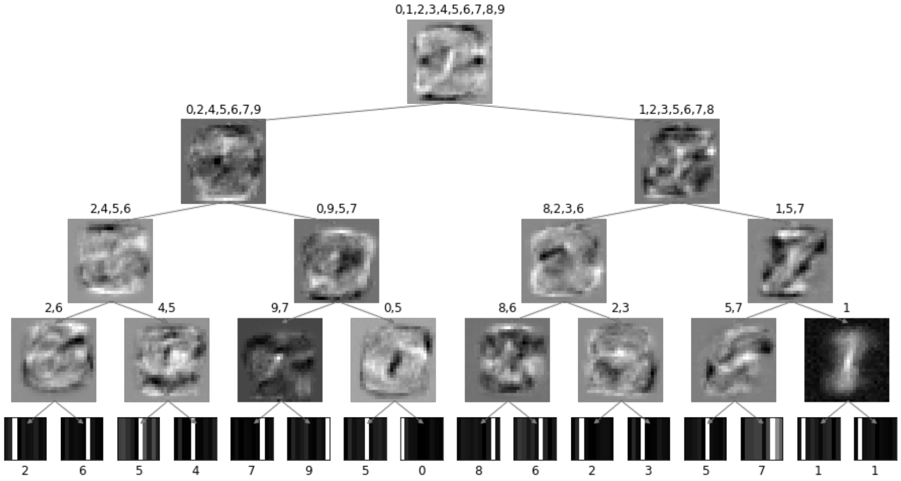


Figure 4.8: Distilled Soft Decision Tree with layer depth of 4 trained on MNIST in 40 epochs. We add labels of the classes with highest probability to each of the nodes within the tree. The squares at the leafs of the tree depict the probability distribution across each of the classes (0-9) using the grey scale.

information about the filter as possible. Note, the filters that split at the top hierarchy of the tree are abstract, whereas nodes later in the tree learn representations that are more easily linked to classes. This aligns with the sequential operation of the neural networks as outlined in Zeiler and Fergus (2014). As a result, upper nodes and the first NN layers must distinguish the majority of classes while layers at the end of the network and leaves at the bottom of the tree can be specialised to focus on particular classes. This aligns with our idea of classification using Soft Decision Trees, because upper nodes have to split most of the classes while lower nodes are more specifically fine-tuned towards the true labels.

Based on a standard MNIST Convolutional Neural Network architecture from the Keras library (Chollet, 2015), we are able to achieve an accuracy of 99% on the test set. Subsequently, we distill the knowledge into the Soft Decision Tree, by training it on the soft targets from the Neural Network. Compared to the 91% accuracy of the distilled 4-layer SDT, the 4-layer SDT exclusively trained on data achieves only 81% accuracy.

To illustrate the classification hierarchy of the tree, the nodes in Figure 4.8 and Figure 4.9 have been attributed to classes with the highest probability. Moreover, the leaves at the bottom of the graph correspond to the class probabilities, to which we apply the *argmax* function for classification (as indicated by the label numbers attributed to each leaf). Using soft decision trees, all classes will be associated with probabilities at every leaf, however in our variation only the leaf with the greatest

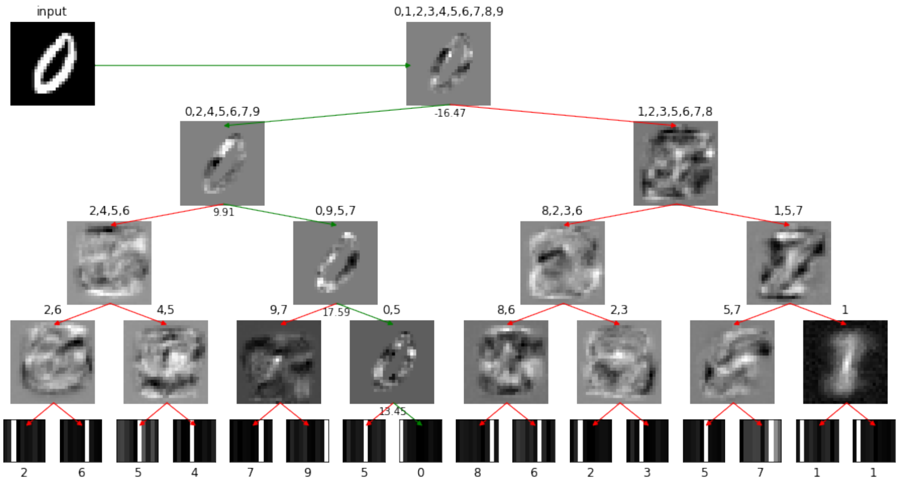


Figure 4.9: Distilled Soft Decision Tree with layer depth of 4 trained on MNIST illustrating a classification of the digit zero. We add a path with highest probability in green to retrace the decision to a sequence of filters.

level of certainty is responsible for the result. We highlight the most probable classes in decreasing order at each node to facilitate interpretability.

The decision trace in Figure 4.9 illustrates the path associated with the highest probability leaf of a specific instance. A green arrow indicates the path with the highest probability, responsible for assigning the final prediction. On the green path, the numbers below each inner node are the pre-activation logit $\beta(xw_i + b_i)$, where we scale β the correlation of input x to each given mask w_i and add a bias b_i. Afterwards we apply the sigmoid (σ) activation function which breaks at 0. A negative correlation leads to a left branch, while a positive one leads to a right branch. Moreover, we present the correlations within each node mask where the homogeneous single-colour area represents zeroes. Positive correlations are denoted by lighter pixels while negative correlations are denoted by darker pixels.

As the tree-depth increases, the accuracy and fidelity of the SDT is improved, but it is at the expense of computational resources and interpretability. The inclusion of additional filters may make it easier to find a sufficient representation of the NN, however the number of nodes becomes too large to be intuitively comprehensible. A four-layer SDT consists of 15 nodes, whereas an eight-layer SDT already has 255 nodes, with each node contributing to the classification to a lesser degree. The masks of each node become less significant as the number of masks increase, thus becoming accumulative and thereby making the classification process more difficult to understand as each node models weaker irregularities.

Tree Depth	Distilled	Accuracy	Fidelity
4	No	81%	–
4	Yes	91%	92%
5	No	91%	–
5	Yes	91%	93%

Table 4.1: Accuracy and fidelity of different Soft Decision Trees on the MNIST dataset with and without Distillation.

4.2.3.2 Fashion MNIST

Label	Description
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

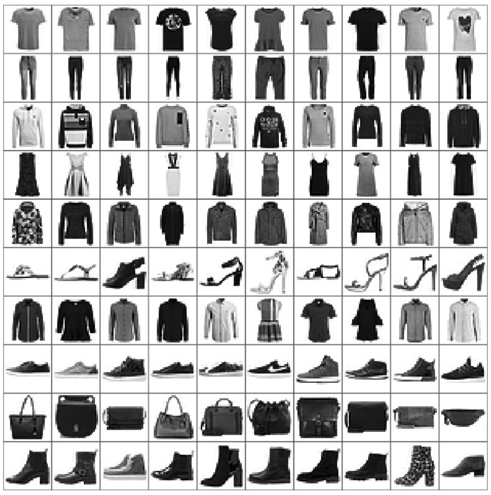


Figure 4.10: Left: Different classes of the fashion-MNIST dataset. Right: Example images of the Fashion-MNIST dataset (Xiao et al., 2017). Each row corresponds to the labels in the table on the left.

As we explore different datasets to assess the adaptability of the SDT distillation method, we gradually increase the difficulty of the domain by varying the complexity of the objects. The fashion MNIST dataset, provided by Xiao et al. (2017), is made up of 70000 samples of different fashion items depicted as 28x28 grayscale images. As the dataset contains three-dimensional objects rather than handwritten digits, it was proposed as a more challenging task than that of MNIST.

As with MNIST, we are expected to classify 10 different classes, as shown in Figure 4.10. The SDT is trained in 60 epochs and a stopping criterion is introduced that halts the optimisation process after 10 consecutive epochs without improvement.

Using a tree with 4 layers, we are able to attain 78% accuracy on the test set through Distillation with 81% fidelity to the network. Prior to this, we train a convolutional neural network in order to identify soft targets for distillation distillation, which yields 93% accuracy on the test set.

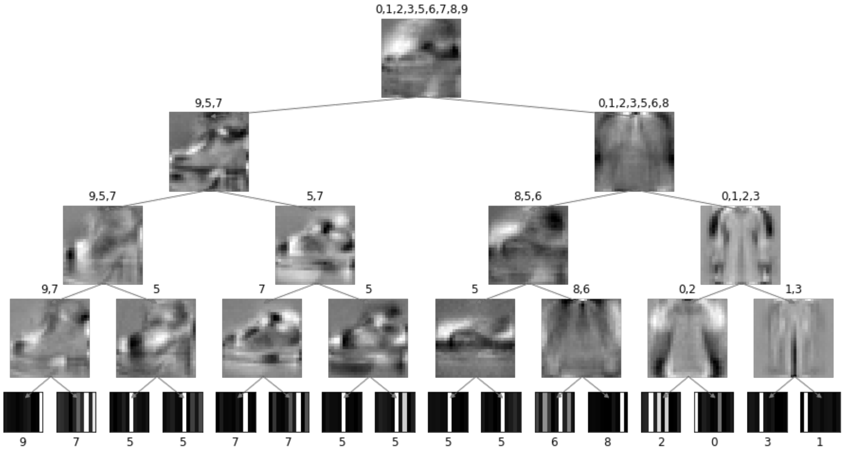


Figure 4.11: Distilled Soft Decision Tree with layer depth of 4 trained on Fashion-MNIST after 40 epochs achieving 78% accuracy. Various filters highlight the distinguishing features of specific clothing items.

Looking at the nodes located in the first layers of the tree, it is evident from the depicted masks that the model has learned to distinguish between clothes on the right and shoes including bags on the left. The significant differences in underlying structure seem to provide a good basis for dividing the classes for classification purposes. This is also consistent with the abstractions of neurons in the first layers of NNs, as demonstrated by Zeiler and Fergus (2014).
However, increasing the depth of the tree does not appear to enhance fidelity or accuracy significantly while reducing the ease of understanding the tree as outlined in Table 4.2.

Tree Depth	**Distilled**	**Accuracy**	**Fidelity**
4	No	72%	–
4	Yes	78%	81%
5	No	82%	–
5	Yes	83%	83%

Table 4.2: Results of different Soft Decision Trees on the Fashion-MNIST dataset with and without Distillation.

4.2.3.3 CIFAR-10

In order to increase the difficulty of the image classification task, we also conducted experiments using the CIFAR-10 dataset (Krizhevsky, 2009) as it presents an even more demanding challenge. This dataset consists of 60000 colour images that belong

Label	Description
0	Airplane
1	Automobile
2	Bird
3	Cat
4	Deer
5	Dog
6	Frog
7	Horse
8	Ship
9	Truck

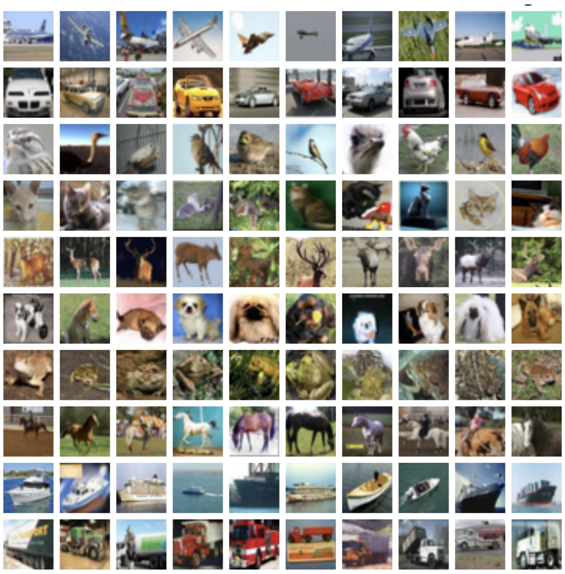


Figure 4.12: Left: Different classes of the CIFAR-10 dataset. Right: Example images of the CIFAR-10 dataset (Krizhevsky, 2009). Each row corresponds to the labels in the table on the left.

to 10 different classes. These images have a size of 32 by 32 pixels with three channels (RGB).

An advantage of convolutional neural networks for these tasks is their ability to learn feature mappings from input images. Because objects can be depicted from many angles and positions, spatial invariance is essential for a CNN to be able to perform well in these types of tasks. Using the Convolution Neural Network without considerable hyperparameter tuning, we achieve a 95% accuracy on the test set.

Following several iterations with different trees of varying size, the best possible achievement we are able to achieve is around 35% test accuracy. The layer depth did not appear to have a significant impact on accuracy, since the model was underperforming regardless of the layer depth. The results showed that varying the layer depth from 4 to 8 resulted in only marginal performance differences (33%-35%). Distilling the information in the Soft Decision Tree of this network did not demonstrate any significant improvements, with performance being improved by less than 1%.

Soft Decision Trees may be limited in their ability to represent complex images and therefore be unable to classify complex images, due to this fundamental limitation. As evidenced by Figure 4.13, the learned filters used to represent the information considered by the classification algorithm are producing incomprehensible abstractions. This type of highly incomprehensible format makes it impossible to reason with the tree and misses the main purpose of the use of the Soft Decision Tree approach. The

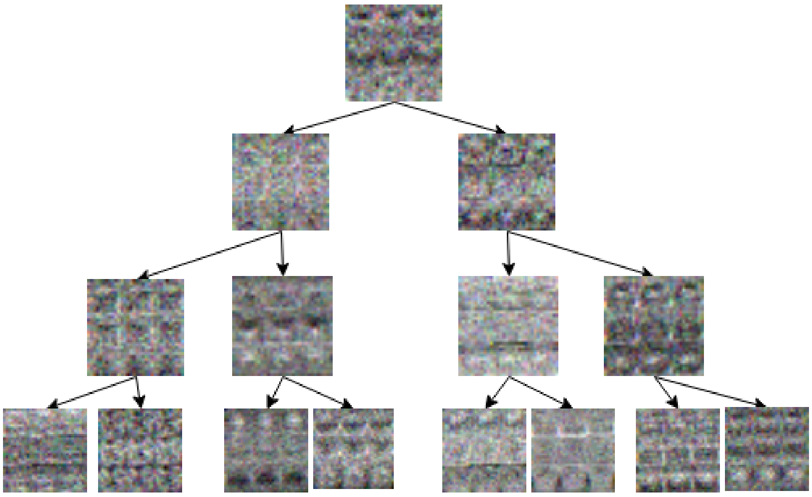


Figure 4.13: Soft Decision Tree with layer depth of 4 trained on CIFAR-10 after 40 epochs.

many different positions of the objects can cause difficulties as we only have a limited amount of nodes and connections to perform effective classification.

4.2.4 Limitations of Soft Decision Trees

In all of our investigations, the results indicate that trees may vary greatly in terms of accuracy and fidelity when it comes to varying the depth of the trees. We conclude that the accuracy-intelligibility trade-off cannot be resolved by Soft Decision Trees, rather this method can be viewed as a means to cover a space between the extremes. Distillation has the advantage of being adaptable to a wide variety of architectural designs and requiring few assumptions. Although such flexibility may be desirable, it may not be accurate to assume that the underlying network is being described as closely as possible. Distillation may allow for the transmission of information to a certain extent, however there are limitations through approximation when utilising only soft targets for the transfer of information. In particular, the utilised node mask does not possess the ability to adapt to complex domains. The lack of feature mappings available to Soft Decision Trees renders them fundamentally ineffective in the application of explaining complex image classifiers. While soft targets may force a closer approximation of the inherent model structure into the extracted representation (see the Fashion-MNIST example), the underlying representations differ significantly. A node-mask filter in the Soft-Decision-Tree is limited to a linear mapping, whereas the feature mappings of Convolutional Neural Networks are based on non-linear convolutional transformations that allow for powerful feature representation (as will be demonstrated in chapter 6 and 7). In addition, the expressiveness of the operations are limited when compared to logic-based methods that would be able to capture full first-order logic. Despite the fact that a soft decision tree representation is fully differentiable, it lacks the precision and power of logic.

The result is that both the extracted representations as well as the connections among these representations impede the attempt to approximate a black-box model as closely as possible. In the next chapter we will create a synthesis of what has been presented and motivate the choice of a neural-symbolic approach.

Chapter 5

Need for Neural-Symbolic integration for XAI

Based on a review of a variety of explainability methods, we have focused on two specific methods that address the pressing problem of opaque autonomous decision-making by focusing either on the inherent representations of the models or on its sequential operations. Post-hoc model-agnostic methods are currently the most widely used, owing to their practicality. As such, these methods provide domain experts with the tools necessary to gain some degree of understanding of the underlying mechanisms that influenced a particular decision.

The representation implied varying degrees of expressiveness based on the data, domain, and algorithmic performance. A logic or tree-based approach may achieve greater transparency with regard to the underlying rules that lead to a decision. In the case of images, this level of detailed understanding may be undesirable as it does not further understanding, as shown by our result.

Nevertheless, with increasing complexity of tasks, it is desirable that an explanation method be flexible in abstraction, use intermediate representations of the model. Ideally, the model should be capable of learning and implementing data transformations that will enable powerful yet comprehensible representations to be generated. It should be noted that, although a hierarchical structure is created in the Soft Decision Tree, the data fed into the nodes at each layer is still the untransformed input vector. Though it helps with comprehension of individual nodes for simple datasets, this negatively impacts the model's ability to learn representations that are able to capture information that is needed to achieve high levels of accuracy and fidelity in complex tasks.

In order to achieve intuitive and powerful model explanations, we may require XAI methods to allow for representations that can capture powerful concept repre-

sentation based on data integrated with comprehensive operations. A powerful yet intuitive approach to XAI may only be possible through both the extraction of inner representation in conjunction with the ability to connect the concepts using logical explanations.
Our study of the TCAV approach revealed that it is effective at extracting representation and subsequently linking the output of the model with high-level concepts. By creating explanations on a concept level, this allows for intuitive explanations, but does not offer the opportunity to grasp the underlying relationships of such concept symbols in the model reasoning process. The study of Soft Decision Trees indicates, however, indicates that methods which allow full transparency into the operational structure of the model may be unable to handle complex tasks.

Currently, most XAI methods are very specialised in the sense that they tend to focus on one aspect of the Machine Learning pipeline rather than considering it as effective communication bridge more broadly. As a result of this, there is another major limitation: Current approaches to XAI are considered to be one-way communication channels. Because of the current design, expert knowledge cannot be incorporated back into the network once we gain insights into the behaviour of the model. As outlined in Bader and Hitzler (2005), we have the capability to extract and revise knowledge, but are currently lacking a mechanism for reintegrating them back into the model. The propositional knowledge extraction method TREPAN is also subject to this limitation. The vast majority of XAI methods focus on obtaining the extracted representations and validating them, opposed to tightly integrating the approach with the original network.

XAI, if viewed as an effective communication bridge, should be capable of feeding information back into the network if we are able to access representations and their relationships accurately. The model and its representation would be continually refined and consolidated over time through interaction with domain experts who can review and revise the information.

5.1 Neural-Symbolic Systems

In the following paragraphs we illustrate why a neural-symbolic approach may be able to overcome some fundamental limitations that have been demonstrated in the experiments above. In addition to taking advantage of model-inherent representations, it permits us to counteract any undesirable model behaviour and take advantage of logic's expressiveness. We will later demonstrate the benefits of such an integration through a variety of experiments.

To begin with, we investigated the hidden representations within a trained Neural Network, but were unable to reason directly about the underlying operations. Without an understanding of the behaviour of the model, the trust and understanding gained with this technique will always remain limited. We then evaluated the approximation of a model by reducing the complexity of its underlying structure, thus allowing for the extraction of knowledge on how the model operates. Nevertheless, we also saw the limitations of the task in terms of its technical complexity, due to the lack of versatility.

In light of the above limitations, we consider the integration of logically connected symbolic representations in combination with the powerful learning abilities of neural networks to be the most promising method of a holistic approach to explainable Artificial Intelligence. There has been a steady increase in research efforts devoted to breaking down the inherently incomprehensible information representation found in neural networks. Comparatively fewer attempts have been focused on rectifying several fundamental shortcomings through the integration of symbol grounding and logical reasoning.

A primary goal of this integration is to retain the ability to learn from data and experience while also being able to reason based on what has been learned. The use of Neural-Symbolic integration has the potential to address a number of notable shortcomings that characterise standard neural networks and their explanations, including limited generalisation, lack of robustness, and lack of comprehensibility. It is noteworthy that there are a number of integration methods that relate to explainability to some degree, but this dimension is often not of primary concern.

Although neural networks and symbolic systems are frequently regarded as two irreconcilable methods, their differences are more subtle than is commonly assumed. Symbolic systems operate at a symbolic level, which involves reasoning on abstract entities in accordance with a logical structure. The purpose of a logical system is to model the same type of reasoning utilised by humans in their everyday lives, thereby rendering the system more comprehensible. It should be noted that there is a difference between human reasoning and humans explaining their reasoning. We intend to emulate the latter using the proposed neural-symbolic XAI method in chapter 6 and to do so, we use elements of symbolic systems that are derived from the former.

Conversely, neural networks operate at a sub-symbolic level. The individual neurons do not necessarily constitute a recognisable concept. A model may rather rely on its predictive capacities rather than any robust abstract reasoning to model weak statistical regularities in a dataset. Thus, integration could connect low-level information processing, such as perception and pattern recognition, with reasoning and explanation at a higher level of conceptual information (Garcez et al., 2008).

It is the complementary nature of symbolism and connectionism that is at the foundation of the Neural-Symbolic approach. It is widely acknowledged that symbolic systems are unable to match the versatility and scalability of learning algorithms used in neural networks (Garcez et al., 2008). However, neural networks lack the ability to generate verifiable and understandable abstract reasoning capabilities that are associated with symbolic systems. When translated symbolically, 'cows give milk' does not refer to any specific cow, but can be generalised to encompass all cows. It is understood that neural networks are limited to the set of training data provided, and that there are no guarantees that 'cow', 'milk', and their relationship will be encoded correctly (Mundt et al., 2020). In fact, research has shown that Neural Networks are susceptible to minor pertubations in low-level input, referred to as adversarial attacks (Kurakin et al., 2017).

Besides the complementary nature of these approaches, a general question about differences arises about where the substantive differences are manifested. Odense (2019) provides extensive formal alignments showing that the primary difference between neural networks and logical systems arises from representational differences.

The benefits of the Neural-Symbolic integration include the fact that the representations are abstract, reusable, and general in nature. In fact, they alleviate some of the key issues associated with current Deep Learning (DL) methods. Although the data efficiency and sample complexity of deep learning systems are highly computationally and data intensive, symbolic approaches are less demanding. In addition, DL approaches do not tend to generalise out of distribution and provide a limited basis for a transfer of knowledge, whereas symbolic representations overcome these limitations. Moreover, DL systems are opaque, in contrast to symbolic approaches that follow a transparent, human-comprehensible decision making process (Garcez et al., 2019).

If the right level of abstraction is applied, such as objects, concepts, and their relationships, explanations based on logical operations might be effective, even in the high-dimensional domains of images and videos. Moreover, a comprehensive generalisation of models may not be possible until a higher conceptual level is attained.

Upon analysing knowledge extraction methods and investigating the advantages of using soft decision trees to extract information into tree-based structures, we concluded that these methods are insightful and usable, but are difficult to apply in safety-critical fields where accurate and reliable information regarding a system's behaviour is needed. A significant contributing factor to this issue is that most explainability methods fail to take into account the decomposition of hidden representation within a network.

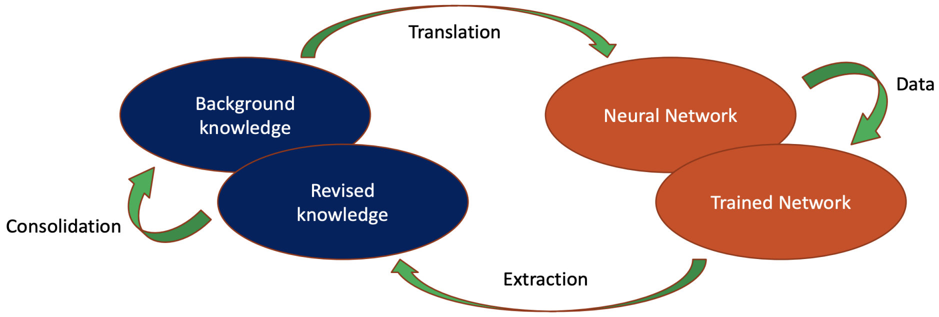


Figure 5.1: Illustration of the neural-symbolic cycle (Garcez et al., 2001): knowledge extraction will be carried out by querying a deep network interactively and learning continually, thus applying the neural-symbolic cycle multiple times. The neural-symbolic cycle can be seen as offering common ground for communication and system interaction, allowing for a human-in-the-loop approach. Symbolic knowledge representations extracted from the learning system at an adequate abstract level for communication with users should allow knowledge consolidation and targeted revision. In the next chapter, we will evaluate its reasoning capability as wee as applicability in the context of improving fairness constraints. Knowledge inserted into the training of the deep network in the form of first-order logic constraints will be shown to improve fairness while maintaining performance of the system in direct comparison with current standard methods on different fairness metrics.

Figure 5.1 illustrates the general principle behind using Neural-Symbolic techniques to approach Machine Learning problems. A system is proposed, where one side consists of a symbolic representation that is both writable and readable by human experts, while the other side comprises a Neural Network with the potential to fully utilise the powerful connectionist training methods. With an iterative loop, symbolic (expert) knowledge can be embedded into the Neural Network as well as learned and refined knowledge extracted from the Neural Network. Using raw data, pre-defined rules can be adjusted, allowing them to be modified as needed.

Systems that synthesize neural networks and symbolic representations typically address each of the following characteristics explicitly: Representation, Extraction, Reasoning, and Learning.

Knowledge representation can facilitate the mapping between the integrated symbolism and connectionism. There are various types of representations, such as rule-based, formula-based, and embeddings. In the manner described previously, the objective may be to extract symbolic information from a trained neural network for explaining and reasoning purposes. Furthermore, efforts have been made to facilitate the process of learning from data directly into a neural-symbolic systems. For instance, Inductive Logic Programming (ILP) can be used to develop logic programs directly from examples (França et al., 2014). Further, learning with logical constraints has generally proven beneficial for improving data efficiency (Garnelo and Shanahan, 2019). These constraints may be implemented, for example, as a logic module layered on top of a regular neural network. Consequently, models are able to gain insight into relations in-between the abstractions as well as assist in guiding the model towards explanations. Also, one of the primary functions of neural-symbolic systems is to facilitate reasoning. As an example, the system might facilitate the generation of new knowledge based on the knowledge acquired during the training phase (Garcez et al., 2002).

Moreover, explainability is becoming an increasingly important characteristic of neural-symbolic systems. Neural networks, due to the extensive number of parameters, provide powerful learning capabilities when the number of hidden units, number of layers, as well as optimisation and regularisation techniques are correctly set up. As models are being utilised in areas with impactful automated decision making, there is an increasing need for explainability and interpretability in neural-symbolic systems. Early on, several proposals had been put forward to extract logical rules from neural networks (Craven, 1996). Despite these efforts, most methods were unable to cope with the exponentially increasing complexity of the data and network.

Ideally, approaches to explainability in AI should facilitate explanations that are consistent with human explanations via symbolic reasoning and are therefore intuitive to understand. However, this should not imply that the model capabilities are fundamentally simplistic and susceptible to under-fitting to complex scenarios. Our goal is to utilise the power of learning through differentiable optimisation over distributed representations to provide task solutions from data-driven models in conjunction with an interface that allows users to reason about what has been learned.

At present, neural networks that are sufficiently trained may contain concept representations that are human-comprehensible and have an inherent logic. However, there is no method that is capable of precisely obtaining information of this type. In the current state of Knowledge Extraction, techniques only approximate the input-output behaviour, and we have no way to compel the extracted representation to follow the underlying logic of the network. A substantial share of explanation methods have traditionally been achieved through post-hoc explanation techniques (as outlined in the taxonomy). Even though an explanation that is solely based on statistical techniques may be beneficial, it is far from being a guaranteed and credible explanation. The majority of explainability methods are not powerful enough to provide guarantees about the truthfulness or accuracy of the explanations in relation to the underlying models, and current metrics lack an effective method of expressing the uncertainty associated with these models. The measured fidelity is supposed to be a reliable indicator of the conformance of our representation to the underlying model. It has been demonstrated, however, that this metric may not be powerful enough to find representations that allow the user to understand the reasoning behind a given representation (White and Garcez, 2020), a gap that symbolic integration could potentially address.

In the process of providing explanations, one of the limitations of current post-hoc XAI methods is that they do not consider domain knowledge and general background information.

5.2 Reasoning about What Has Been Learned

Following from this analysis, we propose to define XAI more broadly as the alignment of model behaviour with human comprehension. The presented approaches have employed different methods for conveying insight into the model itself, but the most essential desiderata remains the communication between the model and the user.

Therefore, if we wish for intuitive human-like explanations as the main communication method, the alignment must take place at some higher level of abstraction. While the predominant low-level and statistical-based explanations are effective for debugging models, logical and reasoning-based explanations require more abstract grounded knowledge utilising higher-level concepts.

Moreover, relying on approximation by any explanation method necessarily results in a loss of information and, consequently, may reduce the quality of the explanation. In contrast, a strong alignment could yield more accurate information if the fundamental building blocks are the same.

It appears reasonable to exert a common ground of inherent logic from the ground up in order to facilitate mutual interactions between humans and artificial intelligence systems. This common ground can be conceptualised as the modularity that integrates perception at the sub-symbolic level with reasoning at the symbolic level. AI advances have provided robust solutions to a number of perception-related problems. Nevertheless, we would like to facilitate the integration of logic using symbolic representations in order to ensure that the model can be understood at a fundamental and intuitive level.

In the following chapter, we propose a framework that allows for a direct interpretation of the abstract representation and operations in a neural network. Ideally, this functionality would be imposed after training so that it could easily be incorporated into existing applications. Furthermore, to guarantee that the structure is able to adapt to the complexity of tasks, considerable flexibility will be needed. If we wish to ensure that effective communication is not compromised by irreconcilable differences, we should ensure that the model is accessible by means of human-interpretable operations and abstractions that are accurate.

By using this approach, we anticipate to overcome some of the fundamental limitations we observed in the Soft Decision Tree experiments. These limitations include the limited ability to solve complex tasks while simultaneously understanding the model's reasoning process comprehensively. Furthermore, unlike the TCAV demonstration, the proposed method may give us a better understanding of operational reasoning capabilities using concept representations within the model.

In addition, Neural-Symbolic integration may allow us to overcome the static limitations of the current machine learning paradigm. Explainability methods of today do not provide us with the ability to act on extracted knowledge. When undesired properties are discovered, the only means to influence the model is to retrain it until a satisfactory model is found. However, retraining as a process is unguided and can only be influenced indirectly through the collection of additional data. The result is that many explanation methods are only limited in their usefulness, since the consequences commonly involve catastrophic forgetting of acquired information (French, 1999; Parisi et al., 2019). This refers to the phenomenon that previously learned information is erased upon learning new information. In the neural-symbolic approach, the goal is to allow for interactive continual reasoning about what has been learned in order to improve the model as needed.

It is important to distinguish between reasoning as it occurs during learning, which is predominantly concerned with how information is used to accomplish a task, and reasoning as it occurs after we have learned something. It is the latter question that is fundamental to explainability whereas the former question represents the central question addressed by a wide variety of model architectures and optimisation methods. In the following chapter we will address both, and we strive to produce mechanisms that allow us to reason about information that accumulates inside the model during the learning process. Nevertheless, both perspectives are inherently linked as the model reasoning steps during optimisation are integral to how we conduct the reasoning process regarding what hase been learning.

We are going to investigate the potential of the extended Logic Tensor Networks (LTN) framework (Serafini and Garcez, 2016) to enable the full Neural-Symbolic circle, including knowledge extraction and translation as shown in Figure 5.1. It would demonstrate the potential of Neural-Symbolic systems in enabling bi-directional communication between the AI system and human experts and users and, therefore, make a significant contribution towards creating an understandable, continual, and interactive artificial intelligence system.

Chapter 6

Neural-Symbolic Integration for interactive XAI

Futia and Vetrò (2020) underline the importance of neural-symbolic integration for explainability by suggesting that traditional explainable AI (XAI) methods lack the ability to provide explanations for the variety of target audiences. While most of the explainability methods may be valuable at providing insight to ML experts, domain experts in applications such as finance or healthcare may struggle to interpret the explanations given. It is proposed that *interactive integration with semantically-rich representations is key to refining explanations targeted at different stakeholders* (Futia and Vetrò, 2020). Indeed, having flexibility in the abstract representation of information that forms an explanation is key to leveraging domain expertise through interaction and revision of the decision making process.

We propose to define explainable AI as the *alignment of model behaviour with human values achieved through model comprehensibility and revision using a communication bridge.* With the use of the neural-symbolic cycle (Fig.5.1), we seek to bridge low-level information processing such as perception and pattern recognition with reasoning and explanation at a higher-level of abstract knowledge. If one wishes to obtain intuitive, human-like explanations, the alignment must take place at a high level of abstraction with an ability to drill down to deeper explanations as the need arises, as in the case of a child's sequence of *why* questions. While the predominant low-level and statistical explanations are effective for debugging, logical and reasoning-based explanations require a more abstract knowledge representation utilising higher-level concepts.

Our aim is to query the neural network for symbolic knowledge so that a direct interpretation of abstract representations and operations on those representations becomes feasible. The approach seeks to explain and possibly revise a decision making process post-hoc, and to be model-agnostic. Furthermore, to ensure that the model can adapt to the complexity of tasks in the usual way as popularised by current ML, we aim to retain the advantages of gradient-based, end-to-end learning. As a means of ensuring that the common *communication layer* is not hindered by irreconcilable disparities between the symbolic (discrete) and neural (continuous) representations, we will need to ensure that the model can be queried with human-interpretable operations at an adequate level of abstraction.

The Neural-Symbolic bridge between logic and Neural Network serves as this communication layer, as logic provides the semantic precision required of the questions to be put to the model. Although the use of logic may seem to be a barrier at first sight, it is required to formalise knowledge and interaction with a precise semantics. The gap that may exist can, in principle, be filled by building wrappers to formulate logical queries e.g. using natural language (Singh et al., 2020).

The core building block of the system will be the usual logical operators (conjunction, disjunction, negation, implication). The goal is to provide an intuition into the operations that the ML model has inferred, based on the observed data of a given task. The logical operators *connect* the symbol representations also in the usual way. Symbols are tangible references which will be used to denote abstract concepts that arise through learning of model-specific, data-driven representations. These abstract concepts will be derived from within a trained model, giving rise to explanations that are grounded on the model's inherent representation and operations.

The neural-symbolic framework adopted in this chapter is that of Logic Tensor Networks (LTN) as described in Serafini and Garcez (2016) and Badreddine et al. (2020). However, instead of treating the learning of the parameters from data and knowledge as a single process, we emphasise the dynamic and flexible nature of training from data, followed by querying the trained model for knowledge and then by consolidating that knowledge in the form of constraints for further training, as part of a cycle with stopping criteria defined by the user. We make LTN iterative by saving the parametrization learned at each cycle in our implementation, while not requiring the use of any specific model architecture, thus making the approach fully model-agnostic.

By using LTNs, we wish to incorporate representational capabilities with greater complexity, due to the addition of information-rich latent representations. However, we retain the ability to incorporate inherent logic through the symbolic interpretation.

We will begin this chapter by introducing the LTN framework, which we will employ and adapt for the purpose of continual learning and reasoning about learned knowledge. Additionally, this involves the specification of different approaches that can be used to acquire knowledge by using the framework. Furthermore, we explore the practical reasoning capabilities more generally of neural networks as part of differentiable fuzzy logic systems. Methods that use fuzzy logic operators to distill logical knowledge into a model architecture employing constraint-based differentiable continuous function optimization are referred to as differentiable fuzzy logic methods. In this study, we will examine limitations associated with neural-symbolic frameworks arising from the difference between the assumptions underlying logical systems and machine learning techniques. To substantiate this, we provide a comparison with a study conducted by Bianchi and Hitzler (2019), improving on the reasoning abilities baseline by considering the outlined considerations. Following these investigations on reasoning capabilities, we will apply this framework more practically within the context of quantitative fairness in order to illustrate how the continual reasoning facilitated by the neural-symbolic approach is capable of guiding the model operations in accordance with human values. As an additional benefit, it will connect to existing XAI methods of the post-hoc category outlined in the taxonomy chapter. We explore how such a method, in this case SHAP, might be incorporated into the framework for interactive revision. Through the use of this integration, we will show that the framework can also complement existing XAI methods by providing the capability to act on the extracted information. Finally, we will illustrate how the ability to incorporate concept groundings into the proposed framework can give rise to powerful logical explanations of the model that are intuitively understandable. The study will describe how concept activation vectors, introduced in chapter 4, can be used to integrate abstract representation into first-order logic. We will demonstrate the implementation and use of this methodology on a variety of datasets and deep learning model architectures. These model architectures have proven to be capable of learning generalisable concepts that are suitable of being disseminated to new and different tasks.

The purpose of this chapter is to demonstrate that neural-symbolic integration can serve as a strong foundation for more trustworthy, fair, and transparent Artificial Intelligence systems. Following a discussion of potential reasoning limitations of fuzzy logic systems such as the one employed in this chapter, we show how the method lends itself to the alignment of human values (such as fairness) with learning systems. The approach provides a means to reason about what has been learned in an accessible manner.

The contributions of this chapter are:

- We implement and outline the use of LTN for continual learning and iterative querying by caching the learned representations and by using network querying in first-order logic to check for knowledge learned by the deep neural network.
- We evaluate the reasoning capabilities of the proposed framework and specify what constitutes valuable background knowledge that contributes to improved deduction.
- We introduce a method that allows one to act on information extracted by any XAI approach in order to prevent the learning model from learning unwanted behaviour or bias discovered by the XAI approach. We demonstrate how our method can leverage an existing XAI method, known as SHAP (Lundberg and Lee, 2017), to discover and address undesired model behaviour.
- We apply the proposed method and tool to the field of quantitative fairness in finance. Experimental results reported in this section show comparable or improved accuracy across three datasets while achieving fairness based on two fairness metrics, including a 7.1% average increase in accuracy in comparison with a state-of-the-art neural network-based approach (Padala and Gujar, 2020).

6.1 Related Work

A variety of methods have been developed to integrate neural and symbolic approaches. Marra et al. (2021) provide a taxonomy of various Neural-Symbolic integration methods and draw connections between these methods and the closely related domain of Statistical Relational Artificial Intelligence (StarAI). For a comprehensive overview of the broad field, this survey provides an in-depth assessment. Our research will focus on the category of Differentiable Fuzzy Logics (DFL), which has been proposed as a category of Neural-Symbolic methods in Krieken et al. (2020). The comparable methods considered here are also computational models that integrate logical reasoning and deep learning into a fully differentiable end-to-end optimizable architecture.

Based on neural-symbolic approaches that employ fuzzy logic with real-valued logical operators, it is possible to construct a differentiable loss function that is derived from

background knowledge, which can involve multiple logical formulas. Through the use of data, these approaches use fuzzy logic to determine the truth value of statements in the form of logical formulas. As the method works in a semi-supervised setting, the data itself can be labelled, unlabelled or sparsely labelled. A representation of this approach can be found in: Badreddine et al. (2020); Serafini and Garcez (2016); Guo et al. (2016); Diligenti et al. (2017); Marra et al. (2019).
In the following, we will provide a high-level overview of the DFL methods before they are discussed in more detail using parts of the LTN framework below. The semantics in Differentiable Fuzzy Logic (DFL) methods are derived from vector embeddings and functions, with logic terms being interpreted in that vector space derived from a real (valued) world. *Objects* are d-dimensional vectors of real values in the domain. *Predicates* are mappings from these vectors to fuzzy truth-values in the interval $[0, 1]$, which may be modelled by trainable NNs that contain learnable parameters θ, considered interpretations, as they assign meaning to the symbols of the logical language. Therefore, varying values of θ will have different implications for the embedded interpretation. For the purpose of emphasising that symbols are interpreted based on their grounding onto real numbers, we will use the symbol *grounding*, denoted by $\mathcal{G}$. The truth values of logical formulas containing quantifiers are determined by using an aggregation function $A : [0, 1]^n \rightarrow [0, 1]$. Moreover, to derive fuzzy truth values for atomic formulas that will be aggregated into the satisfiability of the knowledgebase of logical formulas, $\mathrm{Sat}(\mathcal{K})$, DFL methods rely on fuzzy logic operators that preserve (some) properties of Boolean logic (Krieken et al., 2020). In LTN and other DFL methods, the parameters θ, associated with the predicate groundings, are learned using the maximum satisfiability with respect to the knowledge-base $\langle \mathcal{K}, \mathcal{G}_{\boldsymbol{\theta}} \rangle$ as an optimisation method. This is subsequently reduced to an optimisation problem that can be approached using gradient-descent methods:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\arg\max}\ \mathrm{Sat}_A(\mathcal{G}_\theta(\mathcal{K}))$$

subject to an aggregation A of all logical formulas. $\boldsymbol{\theta}^*$ is obtained by minimising a loss function that encompasses grounded knowledge $\langle \mathcal{K}, \mathcal{G}_{\boldsymbol{\theta}} \rangle$ and maximising satisfiability. Upon convergence of the satisfiability to 1, the network satisfies all formulas contained in the knowledge base $\mathcal{K}$. In light of the fact that all fuzzy operators, including aggregations, are differentiable, gradient descent can be employed to achieve this optimisation.

In order to contextualise the method employed, we would like to highlight two related approaches in particular. Hu et al. (2016) present a framework that utilises a distillation process, which we describe in the section on soft decision trees, to learn from data and rules. The purpose of the distillation method using soft decision trees is to provide a simplified model for enhancing explainability, whereas in the approach provided by Hu et al. (2016), the authors employ this method for translating logical formulas into a neural network. In this scenario, a neural network is used as a student model and a teacher model for distillation. Essentially, the student network is used for inference, while the teacher encodes logical rules into the network. It is done by combining the losses from the student and teacher networks, which is a convex combination of each loss. Accordingly, the student network is iteratively trained to encode a set of constraint clauses. However, the approach to encoding logical constraints into the network is different from DFL methods.

In the LTN method, constraints are directly imposed on the network parameters during the learning steps, and the satisfiability of each formula is explicitly optimised. Through direct connection between the logical formulas and the network, the approach allows for more transparent representation of individual constraint satisfiability as well as higher assurance of explicit behaviour.

Marra et al. (2019) introduce an approach called LYRICS, which is closely related to the LTN framework which will be described in detail below. As with other Differentiable Fuzzy Logic approaches, logical constraints are translated into a loss function that measures the overall statisfiability of the knowledge-base based on the parameters of the network. The essential difference the authors propose is the addition of a hyper-parameter, namely weights to each of the logical formulas. As will be shown in the section on modularity of loss, this addition can be easily accomplished in the LTN framework in a variety of ways and is therefore not a disavantage of the framework itself.

While the LTN approach enables a combined learning of logic and data, Deep Logic Models (DLM) (Marra et al., 2020) and KENN (Daniele and Serafini, 2019) serve to impose logic onto the model during inference. Both approaches are based on the modification of the model prediction, in accordance with a set of logical constraints. As part of the KENN framework, a knowledge enhancer is a function that modifies the output of a neural network according to a set of weighted constraints, formulated in quantifiable terms Daniele and Serafini (2019). A training set provides the basis for learning the weighting of the logical clause constraint. The LTN framework presents training data and logical constraints uniformly based on logic formulas, therefore ensuring that the data and knowledge are consistent. KENN, on the other hand, may minimise the loss by overlooking logical clauses, if the data is preferred to the knowledge. As opposed to LTN, where the constraints may be written in full first-order logic, KENN is limited to universally quantified clauses. DLM differs from KENN primarily in the way constraints are imposed on the predictions (Marra et al., 2020). An undirected graphical model is used alongside the neural network in this case, in which the graphical model corresponds to the logical constraint. As a result, they correspond to groundings of propositional formulas based on the target truth assignment and the weights of the formula. Again, the weights of each formula are learned through optimisation. Similar to LTN, DLM evaluates constraints with fuzzy semantics. However, it only takes into account propositional connectives, whereas LTN supports both universal and existential quantifiers.

The approach of Markov Logic Networks (MLN) and its variants differs from the method of real logic adopted in this context. MLNs measure the degree of truth of a formula according to the number of models that satisfy that formula; i.e. the more models that satisfy it, the higher the degree of truth (Richardson and Domingos, 2006). Hybrid MLNs introduce a relation between real features and constants, which is pre-determined rather than learned (Jue and Domingos, 2008). Instead, real logic determines the level of truth of complex formulas through (fuzzy) logical reasoning, and the connection between the features of different objects is learned through optimisation. Furthermore, MLNs are based on a closed world assumption, while Real Logic operates under an open-world assumption.

The use of Logic Neural Networks (Riegel et al., 2020) facilitates a more modular approach to the architecture of the network, since it corresponds directly to a system of logical formulas. Each individual neuron is associated with an element of a formula in real-valued logic, resulting in disentangled representations. Each of the formulas here is accompanied by a learnable weight as is also done in LYRICS (Marra et al., 2019).

This architecture results in a representation that is inherently interpretable since individual neurons function as logical connections that can be understood. In contrast, we use distributed representations, which are learned to follow the specified real-valued logic constraints. This allows for a greater degree of flexibility in learning as well as the use of low-level input data, such as images. In spite of the fact that distributed representations may emphasise functionality over inherent interpretability, we will show how our approach can address this challenge by grounding the learned representation in human-comprehensible symbols.

6.2 Method

The specific framework used in this section builds upon Logic Tensor Networks (Serafini and Garcez, 2016). Hence, we utilise its central notion of *Real Logic* as a fully-differentiable first-order logic. However, our approach is modified by instead of treating parameter learning using data and knowledge as a single process, we emphasise the dynamic and flexible nature of the process of training from data, querying the trained model for knowledge, and adding knowledge in the form of constraints for further training, as part of an interactive cycle (as illustrated in Figure 6.2). Therefore, our approach focuses on the core of the LTN method: constraint-based learning from data and first-order logic knowledge. Moreover, we make the framework iterative by saving the learned parametrisation at each cycle in our implementation, while removing the previously proposed use of Neural Tensor Networks for predicate mappings (Socher et al., 2013; Serafini and Garcez, 2016). As part of the experiments below, we utilise a variety of neural network architectures, including standard feed-forward networks when not specified otherwise.

As a result, we refer to the LTN adaptation as a framework for both indicating the semantics of real logic as well as the translation of task learning from data and knowledge into an optimisation process based on the satisfiability levels of individual clauses. In our adaptation of the framework, we do not refer to LTN as a *model* in the classical sense as it is model-agnostic.

In light of this, the framework and experiments described below cannot be viewed as mere applications of the LTN method. Although we are building on the formalised system outlined in Badreddine et al. (2020) that provides us with the differentiable fuzzy logic setting, the actual implementation extends the existing work on several key points. Among the distinctive characteristics of our approach is the attention paid to the explainability aspect and the introduction of various methods for extracting information to reason about what has been learned. It is true that some form of transparency is commonly recognised as a positive side-effect, however giving primary consideration to this characteristic as a core feature is something not prevalent in the LTN and DFL context in general.

Further, with the focus on enabling human-machine interaction, the continuous application of revision, consolidation, and extraction of knowledge is another key contribution. LTN, described in Serafini and Garcez (2016); Badreddine et al. (2020) is centred around simultaneous learning from background knowledge and data using a neural network architecture, while we are interested in extracting information that

allows for oversight along with interactive continual learning.
As we begin this section, we will provide an overview of the aspects of LTN relevant to this thesis. Further, we will outline how the continuous approach fits into this framework. Moreover, we discuss how the existing optimisation function may be extended to provide the user with greater control over the model behaviour and the resultant model reasoning. The purpose of this is to derive a general definition and investigate reasoning capabilities of LTNs and differentiable fuzzy logic systems in general as part of the subsequent sections. Here we show how different underlying assumptions may lead to varying deductive capabilities. This behaviour is examined in more detail, as a comprehensive understanding of the reasoning capabilities of the model and of its potential limitations is fundamental to the development of an interactive framework. In order to demonstrate how the framework can be used in a variety of practical XAI settings, it is necessary to clarify how a neural-symbolic model, trained on data and background knowledge, is capable of deducing knowledge from given information. We can thus gain an understanding of what will constitute adequate background knowledge to construct a system whose reasoning capacity is worthy of investigation. In the absence of reasoning capabilities within the model, an extraction and revision of inherent knowledge would be unnecessary. As we illustrate different levels of deductive capability using different settings of background knowledge and model architecture through an illustrative example of a particular logical axiom, we will expand these findings by assessing the deductive capacity of our approach across two datasets in comparison with existing research from Bianchi and Hitzler (2019). In this study, we demonstrate that, by considering the underlying assumptions and limitations of DFL systems, practical reasoning capabilities can be improved using the continual framework.

6.2.1 Language

Logic Tensor Networks (Serafini and Garcez, 2016; Donadello et al., 2017; Badreddine et al., 2020) implement a many-valued first-order logic (FOL) language $\mathcal{L}$, which consists of a set of constants $\mathcal{C}$, variables $\mathcal{X}$, function symbols $\mathcal{F}$ and predicate symbols $\mathcal{P}$. Logical formulas in $\mathcal{L}$ allow for the specification of background knowledge pertinent to the task at hand. The syntax in LTN follows from FOL, with formulas consisting of predicate symbols with negation ($\neg$), binary connectives: conjunction, disjunction, implication, bi-implication ($\wedge, \vee, \rightarrow, \leftrightarrow$) and quantifiers: universal and existantial ($\forall, \exists$). Formulas in $\mathcal{L}$ also facilitate the specification of relational knowledge using variables. As an example, consider the atomic formula $\text{partOf}\,(o_1, o_2)$, which indicates that object o_1 is a part of object o_2. Additionally, since we are interested in learning and reasoning in real-world scenarios, exceptions to the rule may occur. Due to the fuzzy semantics adopted by the language, formulas may be partially true in case of exceptions.

To obtain a broader description of the underlying syntax and semantics, we refer the reader to Badreddine et al. (2020) who provides an extensive definition of the underlying real-valued (fuzzy) logic. Furthermore, Fagin et al. (2021) discusses the correctness and power of real-valued logics more specifically by providing an axiomatisation and subsequently showing what information can be inferred. We confine the methodology in this thesis to the components that are relevant to the framework used and its applications.

6.2.2 Grounding

As for the semantics of $\mathcal{L}$, LTN deviates from the standard abstract semantics of FOL and proposes a concrete semantics where domains are interpreted in the real field $\mathbb{R}$ as defined in *Real Logic* Serafini and Garcez (2016). LTN adopts the term *grounding*, denoted by $\mathcal{G}$, instead of interpretation, in order to indicate that symbols are interpreted in accordance with their groundings onto real numbers. Every object denoted by a constant, variable or term is grounded onto a tensor of real numbers. Function symbols are grounded as functions in the vector space, that is, an m-ary function maps m vectors of real numbers to one vector of real numbers. Predicates are grounded as functions that map inputs onto the interval $[0, 1]$ representing the predicate's degree of truth. Any sentence formulated as a first-order logic expression may be grounded in $\mathcal{L}$ using logical connectives.

The semantics for the connectives is defined according to fuzzy logic semantics: conjunctions are approximated by t-norms (e.g. $min(a, b)$), disjunctions by t-conorms (e.g. $max(a, b)$), negation by fuzzy negation (e.g. $1 - a$) and implication by fuzzy implications (e.g. $max(1 - a, b)$). Readers may refer to van Krieken et al. (2019); Krieken et al. (2020) for a detailed description and analysis of various fuzzy operators and their impact on reasoning.

We list in tables 6.1 & 6.2 a number of these fuzzy logic operators that can be viewed as hyper-parameters within the LTN framework. LTN employs the standard strict and strong negation $N(a) = 1 - a$ in order to ground negated terms.

Krieken et al. (2020) and Badreddine et al. (2020) demonstrate that for the majority of scenarios, the product (Goguen) operators provide a more suitable approach to differentiable learning due to the behaviour of the gradients during the optimisation.

Name	$a \wedge b$	$a \vee b$
Gödel	$\min(a, b)$	$\max(a, b)$
Goguen/Product	$a \cdot b$	$a + b - a \cdot b$
Łukasiewicz	$\max(a + b - 1, 0)$	$\min(a + b, 1)$

Table 6.1: Common conjunction and disjunction operators used in fuzzy logic.

The authors show that multiple fuzzy logic operators can result in exploding or vanishing gradients, and therefore poor performance. Additionally, the operators should propagate the gradients to multiple inputs at once in order to prevent passing updates to a single input.

Name	$I(a, b)$ or $a \rightarrow b$
Kleene-Dienes	$\max(1 - a, b)$
Łukasiewicz	$\min(1 - a + b, 1)$
Reichenbach	$1 - a + a \cdot b$

Table 6.2: Common fuzzy implications.

In general, all clause groundings can be portrayed as a computational graph structure that may incorporate one or more neural networks for the learned projections low-level input to the [0,1] interval. Figure 6.1 illustrates the computation graph for a specific FOL clause utilising fuzzy logic operators. These computation graphs are automatically generated using the LTN implementation accompanying the thesis. Thus, all formulas in $\mathcal{L}$ consisting of constants $\mathcal{C}$, variables $\mathcal{X}$, function symbols $\mathcal{F}$ and predicate symbols $\mathcal{P}$ become differentiable. It is possible to accomplish this through aggregation functions which are a component of the logical language.

An aggregation function determines the semantics of the quantifiers. These aggregations constitute another vital component of the fuzzy logic framework proposed by LTN. In order to obtain a satisfiability level (scalar) that serves as the optimisation criterion for training, it is necessary to aggregate all individual formulas in the knowledge-base $\mathcal{K}$. In other words, each formula will be assigned a fuzzy truth value, enabling further understanding of the statisfiability of each logical clause. The subsequent aggregation may be as simple as taking the average or tailored to impose a particular behaviour on the system (as will be discussed in the section 6.2.5).
Furthermore, universal and existential quantifiers need to be expressed in fuzzy logic in $\mathcal{L}$. They permit statements to specify how many individuals or objects in the domain satisfy an open formula by relating statements to free or bounded variables $\mathcal{X}$. The universal quantifier $\forall$ in the term $\forall x : T(x)$ indicates that everything in the

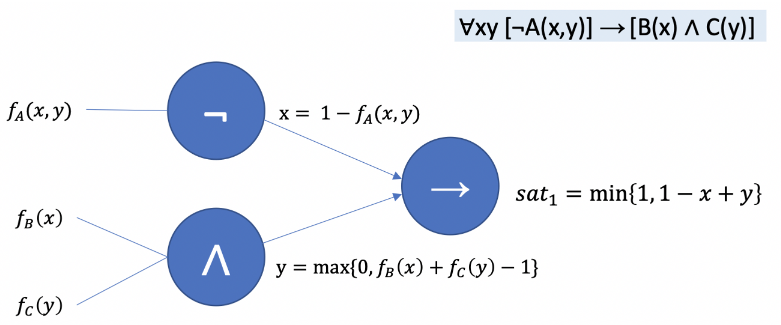

∀xy [¬A(x,y)] → [B(x) ∧ C(y)]
$f_A(x,y)$
¬
$x = 1 - f_A(x,y)$
$f_B(x)$
$f_C(y)$
∧
→
$sat_1 = \min\{1, 1 - x + y\}$
$y = \max\{0, f_B(x) + f_C(y) - 1\}$

Figure 6.1: Diagram illustrating how fuzzy Lukasiewicz connectives and fuzzy implications can be used to ground a clause. Specifically, the following fuzzy logic clauses is grounded in fuzzy logic: $\forall x, y : \neg A(x, y) \rightarrow B(x) \wedge C(y)$. A in this illustration is a binary predicate, while B and C are unary predicates. Furthermore, here x and y represent free variables.

domain satisfies the property denoted by T. Similarly, the existential quantifier $\exists$ in the term $\exists x : T(x)$ indicates that there is at least one element in the domain which satisfies the property T.

Using all available data, one could, for example, enforce or learn about the general behaviour of a model using the universal operator in the LTN framework to evaluate the truth value of particular terms. LTN therefore replaces x with all objects x_i ($x_i \in x$) in the data whenever a quantified free variable ($\forall x$) is part of a formula in $\mathcal{L}$. The truth-value of such formula is then obtained by aggregating the truth-values of each grounded object x_i.

Within the LTN framework, the following are common aggregation functions used for quantification and aggregation of logical clauses Badreddine et al. (2020):

$$A_M(x_1, \ldots, x_n) = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{mean}$$

$$A_{pM}(x_1, \ldots, x_n) = \left(\frac{1}{n}\sum_{i=1}^{n} x_i^p\right)^{\frac{1}{p}} \tag{p-mean}$$

$$A_{pME}(x_1, \ldots, x_n) = 1 - \left(\frac{1}{n}\sum_{i=1}^{n} (1 - x_i)^p\right)^{\frac{1}{p}} \tag{p-mean error}$$

A_{pM} is the generalized mean, and A_{pME} can be understood as the generalized mean when measured in terms of the errors. In this respect, A_{pME} measures the power of each value's deviation from the ground truth 1. We follow the recommendation by Badreddine et al. (2020) and employ A_{pME} with $p \geq 1$ to approximate the universal quantifier $\forall$, and A_{pM} with $p \geq 1$ for the approximation of the existential quantifier

∃, if not explicitly stated. There are certain values of p that result in special cases of particular aggregators:

$$\lim{}_{p \to +\infty} A_{pM}(x_1, \ldots, x_n) = \max(x_1, \ldots, x_n) \tag{6.1}$$

$$\lim{}_{p \to -\infty} A_{pM}(x_1, \ldots, x_n) = \min(x_1, \ldots, x_n) \tag{6.2}$$

$$\lim{}_{p \to +\infty} A_{pME}(x_1, \ldots, x_n) = \min(x_1, \ldots, x_n) \tag{6.3}$$

$$\lim{}_{p \to -\infty} A_{pME}(x_1, \ldots, x_n) = \max(x_1, \ldots, x_n) \tag{6.4}$$

In a fuzzy logic context, these *continuous* (*min* to *max*) approximators can be useful for fine-tuning the strictness of the various quantified axioms. In some cases, *min* may be an appropriate candidate for ∀, when all objects $o_i \in x$ should be strictly grounded in accordance with the term. As a consequence, the high value of p will result in outliers in x being penalised more harshly during aggregation, thereby having a more direct impact on the optimisation process.

In this thesis, we estimate binary connectives by using the product t-norm and its corresponding t-conorm and Reichenbach implication, if they are not specified differently for individual applications below. Badreddine et al. (2020) demonstrate that this setup is superior due to enhanced gradient propagation for robust learning.

6.2.3 Learning

As part of grounding the knowledge $\mathcal{K}$ with its symbols in data, a set of parameters must be learned: $\boldsymbol{\theta}$. Additionally to LTN functions and predicates being learnable mappings, the objects denoted by LTN constants and variables can also be learnable embeddings. Even though we will demonstrate how the framework can accommodate a wide range of model architectures, we will present here the default architecture that is implemented in the accompanying repository. By using a multilayer perceptron to model each predicate, if not explicitly stated, the parametrisation used in this thesis is:

$$\mathbf{h}^{(1)}(\mathbf{v}) = g^{(1)}\left(\mathbf{v}V_P^{(1)T} + b^{(1)T}\right)$$
$$\mathbf{h}^{(2)}(\mathbf{v}) = g^{(2)}\left(\mathbf{h}^{(1)}(\mathbf{v})V_P^{(2)T} + b^{(2)T}\right)$$
$$\mathcal{G}(P)(\mathbf{v}) = \sigma\left(\mathbf{h}^{(2)}(\mathbf{v})V_P^{(3)T} + b^{(3)T}\right)$$

where each $g^{(l)}$ is an activation function, e.g. ReLU, $V^{(l)}$ is a mxn weight matrix, l corresponds to the layer and $b^{(l)}$ a bias vector; σ denotes the sigmoid activation function which ensures that predicate P is mapped from $\mathbb{R}^{mxn}$ to a truth-value in $[0, 1]$.

Since the grounding of a formula $\mathcal{G}_\theta(\phi)$ denotes the degree of truth of ϕ, one natural training signal is the degree of truth of the formulas in the knowledge-base $\mathcal{K}$. The objective function is therefore to maximise the satisfiability of all formulas in $\mathcal{K}$:

$$\boldsymbol{\theta}^* = \operatorname*{arg\,max}_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \ \mathrm{Sat}_A(\mathcal{G}_\theta(\mathcal{K}))$$

which is subject to an aggregation A of all formulas (e.g. the above-mentioned p-mean). This approach is consistent with that taken by other differentiable fuzzy logic methods mentioned above.

The neural tensor network proposed in Serafini and Garcez (2016) has been replaced with a multilayer perceptron due to its enhanced efficiency and general applicability. However, as will be demonstrated, there are no restrictions as to what type of function or neural network may be employed as the method is agnostic. A sigmoid function may be used to appropriate any mapping by projecting the output into [0,1]. In this thesis, we will demonstrate that even intermediate representations of networks can be mapped onto truth values for the purpose of providing explanations for models. Further, we would like to point out that the initial proposition of utilising one neural network per predicate mapping by Serafini and Garcez (2016) is not being utilised in this thesis. The adaptation will use a single network to map several predicates if the dimension of the inputs for the predicate groundings aligns. In other words, if the predicate grounding is also based on the same real features, we will add an additional output neuron instead of adding another neural network. Unless otherwise stated, we use this singular network representation to create a more parameter efficient representation that enables joint information representation for multiple predicate mappings.

6.2.4 Continuous Querying

One of the main benefits of establishing a grounded theory $\mathcal{T} = (\mathcal{K}, \mathcal{G}_\theta)$ is the ability to conduct queries. These queries examine whether a certain fact ϕ holds true based on the grounded real logic truth values that fall within the interval $[0, 1]$.

LTN inference using first-order logic clauses is not a post-hoc explanation in the traditional sense. As presented here, we propose that inference should be an integral part of an iterative method that allows incremental explanation through the distillation of knowledge guided by data. We achieve this by computing the value of a grounding $\mathcal{G}_\theta(\phi_q)$, given a trained network (set of parameters θ), for a user-defined query ϕ_q.

Specifically, we save and reinstate the learned parameters stored in our adaptation of the LTN implementation. This is done by storing the parameters θ resulting

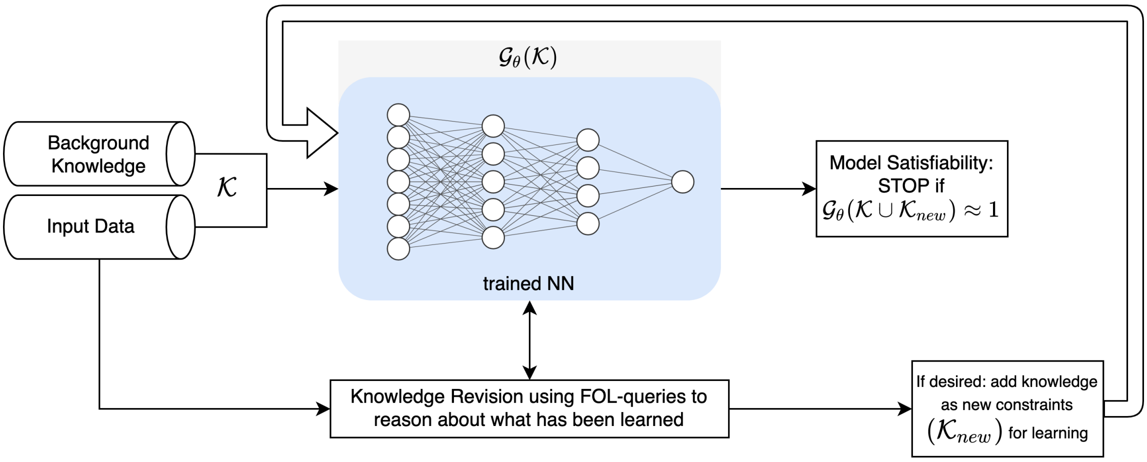

$\mathcal{G}_\theta(\mathcal{K})$
Background Knowledge
Input Data
$\mathcal{K}$
trained NN
Model Satisfiability: STOP if $\mathcal{G}_\theta(\mathcal{K} \cup \mathcal{K}_{new}) \approx 1$
Knowledge Revision using FOL-queries to reason about what has been learned
If desired: add knowledge as new constraints $(\mathcal{K}_{new})$ for learning

Figure 6.2: Illustration of the LTN interactive-learning pipeline: knowledge revision will be carried out by querying a deep network interactively and learning continually, thus applying the neural-symbolic cycle multiple times. The illustration will be modified throughout the thesis to illustrate where exactly the experiments alter the framework.

from the optimisation process of maximising the satisfiability of all formulas in $\mathcal{K}$ $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathrm{Sat}_A \mathcal{G}_\theta(\mathcal{K})$. This also means that changes made to the knowledge-base followed by further training will not reinitialise parameters, but will instead start from saved $\boldsymbol{\theta}^*$. Therefore, as part of the interactive-learning LTN pipeline, it is possible to add knowledge to the previously trained network. Having this functionality allows us to continually query and guide the learning process according to added knowledge $\mathcal{K}_{new}$, an approach akin to that of continual learning (depicted in Figure 6.2). While continual learning is particularly relevant for learning a series of different tasks, our main objective is to explain and refine the existing task.

A query is any logical formula expressed in first-order-logic. Queries are evaluated by calculating the grounding $\mathcal{G}$ of any formula whose predicates are already grounded in the neural network, or even by defining a predicate in terms of existing predicates. For example, the logical formula $\forall x : (A(x) \rightarrow B(x))$ can be evaluated by applying the values of x, obtained from the dataset, to the trained network, obtaining the values of output neurons A and B in [0,1] (corresponding to the truth-values of predicates A and B, respectively), and calculating the implication with the use of the Reichenbach-norm and aggregating for all x using the p-mean.

Algorithm 1: LTN-active learning cycle

```
Input: Dataset, Knowledge (in the form of FOL)
Output: Model satisfiability measured as overall sat-level
for each predicate $P$ in $\mathcal{K}$ do
    Initialize $\mathcal{G}_\theta(P)$    // each $P$ can be a multilayer perceptron or output neuron
for $epoch < num\text{-}epochs$ do
    $\max \mathrm{sat}\, \mathcal{G}_\theta(\mathcal{K})$    // optimize $\theta$ to achieve max satisfiability of $\mathcal{K}$
while Revision do    // user-defined Boolean
    for each FOL-query $\phi_q$ do
        Calculate $\mathcal{G}(\phi_q)$    // query the network to obtain the truth-value of $\phi_q$
        if $\mathcal{G}(\phi_q) < t$    // $t$ in [0,1] is a user-defined minimum sat value
         then
            Add $\phi_q$ to $\mathcal{K}_{new}$
    Apply XAI-method    // we use Shapley values
    for each predicate $P$ do
        Inquire $\mathcal{G}_{\boldsymbol{\theta}}(P)$    // query predicate-specific groundings
        if $\mathcal{G}_{\boldsymbol{\theta}}$ has undesired property $f(\mathcal{G}_{\boldsymbol{\theta}})$ then    // user-determined desiderata
            Revise $f(\mathcal{G})$ to $\mathcal{K}_{new}$    // method-dependent revision
    if $\mathcal{K}_{new} \neq \emptyset$ then
        $\mathcal{K} \leftarrow \mathcal{K} \cup \mathcal{K}_{new}$
        $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{G}_\theta(\mathcal{K})$    // re-train the network
```

Algorithm 1 illustrates the steps we take to continuously refine $\mathcal{K}_{new}$ with a human-in-the-loop. The queries are derived from questions a user might have about the model's response: how does the model behave for a specific group? How does the model behave for particular edge-cases? These questions can be translated relatively easily into FOL-queries. Simultaneously, an XAI method may further inform the user about possible undesired model behaviour which may not be as apparent as the above common questions. This can be accomplished by a variety of XAI methods which may give insight into the functionality of a black box model.

In the section on fairness, the XAI method SHAP (Lundberg and Lee, 2017) will be shown to report a discrepancy in how the variable *reported income* is used by the ML system for men and women, for example. This can be influenced and addressed by adding knowledge to $\mathcal{K}_{new}$ and retraining, as will be illustrated and quantified in the upcoming experiments.

A network revision encompasses the use of new data and new knowledge through training examples and optimisation. In other words, the trainable parameters of the underlying neural network must be adjusted to conform to the revised knowledge base. Along with revising knowledge and continuously updating the system, a user may also introduce new predicates. In a later section of this thesis, we will demonstrate how we can not only revise existing predicate mappings but introduce supplementary predicate mappings at a later stage in the revision process to enhance the system behaviour. Depending on the task, it may be sufficient to add one more output node or to create an entirely new network. Nevertheless, even with a novel network component added to the system, it will be possible to utilise existing learned knowledge through interconnected optimisation in $\mathcal{K}$. A logical formula may utilise logical connectives to establish connections with existing predicate mappings, thereby enabling the optimisation of newly introduced parameters using learned knowledge.

6.2.5 Modularity of data and knowledge

It is an integral part of the learning process to derive new knowledge from given information, thus we want to highlight how this process may manifest differently within the introduced framework.

In Badreddine et al. (2020), the authors illustrate how LTN could be applied to binary classification, multi-class classification, clustering, logical reasoning, and regression. In addition to the ability to apply a Neural-Symbolic method to imitate well-known areas of application, we will explore why both traditional data-driven and logical approaches can be improved by an integrative approach, and how the user may guide the learning process based upon personal preference.

DFL methods raise the question of what constitutes a desirable optimisation signal, when we learn from data and knowledge simultaneously. Usually, the quality of an approach to machine learning is evaluated in terms of its ability to achieve the desired outcomes on held-out data, using metrics such as accuracy, precision, recall, and F1 score.[1] Symbolic reasoning, on the other hand, is typically evaluated according to the validity of its conclusions. These conclusions must, for instance, satisfy certain constraints. It remains an open challenge in an integrated AI system to determine the appropriate evaluation measures and training signals for learning from data and knowledge simultaneously. Throughout this section we will detail the ways in which the user may influence the capabilities of the system in accordance with their preferences and illustrate the extent to which the proposed framework may be adapted to a wide range of training signals as well as improve the performance of purely data-driven or logical approaches.

Using a data-driven approach, we are able to develop a robust mapping of groundings from low-level inputs in domains that are traditionally mediated by symbolic or logical approaches. Thus, real-world data can be processed and the system is less susceptible to failure when observing inputs that are not identical to those previously observed. By being able to interpolate the observed inputs, the ML models have the capability of performing in an open world scenario where new data may be introduced at any time as long as it falls within the sampled data distribution. Many logical systems, on the other hand, are bound to the closed world assumption where knowledge is assumed to be complete. Moreover, supplemental background knowledge can provide an outlet

[1] Precision is the number of true positives divided by the total number of positive results, including incorrectly identified samples. Recall (also known as sensitivity) is the number of true positives divided by the total number of samples that should have been labelled as positives. The F1 score refers to the harmonic mean of the precision and recall.

for overcoming fundamental limitations in data-driven domains where models have been widely used over recent years. Hence, we should distinguish between different forms of knowledge that can be incorporated into an integrated system, which allow for varying types of reasoning that may help overcome existing constraints of purely data-driven approaches.

On the one hand, there are propositional statements in which we state explicitly how an input corresponds to a truth value. Thus, this form of knowledge may be utilised for classification, since the groundings can be mapped according to training data and their respective labels. By explicitly adding propositions to each known mapping, this information can be integrated (e.g. $A(a_i)$ for predicate class A). Alternatively, the user may directly specify input variables of a quantifiable term, ensuring that only certain variables are mapped appropriately (e.g. $\forall a : A(a)$, specifying the range of quantification to variables of a set a or using guarded quantification). The two approaches will be referred to as *positive propositions* for predicates.

As a consequence, *negative propositions* are explicit negative (or counter) examples of predicate mappings. We will demonstrate in this thesis that such propositions are crucial for gradient-based learning of mappings in DFL approaches. Similar to supervised learning environments, such information is required to be accounted for in trained models to learn meaningful mappings that are robust and useful for interpolation.

Notably, under the closed world scenario of particular logical systems, the negative proposition can be derived from the positive proposition due to their complementarity. Hence, for DFL systems applied to closed world domains, such knowledge is required to be derived, as illustrated in the different scenarios of the section to follow.

In contrast to these propositional statements, open quantifiable logic clauses may employ free variables that are utilised to employ general axioms to inform general model behaviour. In these type logical formulas the variable used for quantification is unrestricted since it is the relationship between variables that offers information (e.g. $\forall x : A(x) \to B(x)$). For instance, this allows for quantifiable knowledge to be used to transmit information from areas with data to those with no data. A user may wish to utilise background knowledge to transfer information from a domain in which propositional information is present to another domain in which such information is absent (as illustrated in Figure 6.3).

Furthermore, in a semi-supervised setting, we can supplement incomplete propositions with general knowledge using quantifiable clauses. Even though propositional information in many traditional logic tasks is assumed complete, in real-world situations, it is frequently sampled. The application of quantifiable axioms can be utilised to enrich the model trained on sampled data by guiding interpolation and extrapolation to be more robust.

Moreover, we wish to emphasise that any existing loss function may be incorporated into K. As an example, any loss function $\mathcal{L}_B$ can be expressed as Sat_B-value by declaring that $\text{Sat}_B = 1 - \mathcal{L}_B$. Therefore, users are able to leverage existing classification losses to learn predicate mappings.

Notably, the optimisation signal of DFL methods can be constructed in a modular fashion, providing further benefits. An individual may elect to adopt either a data-

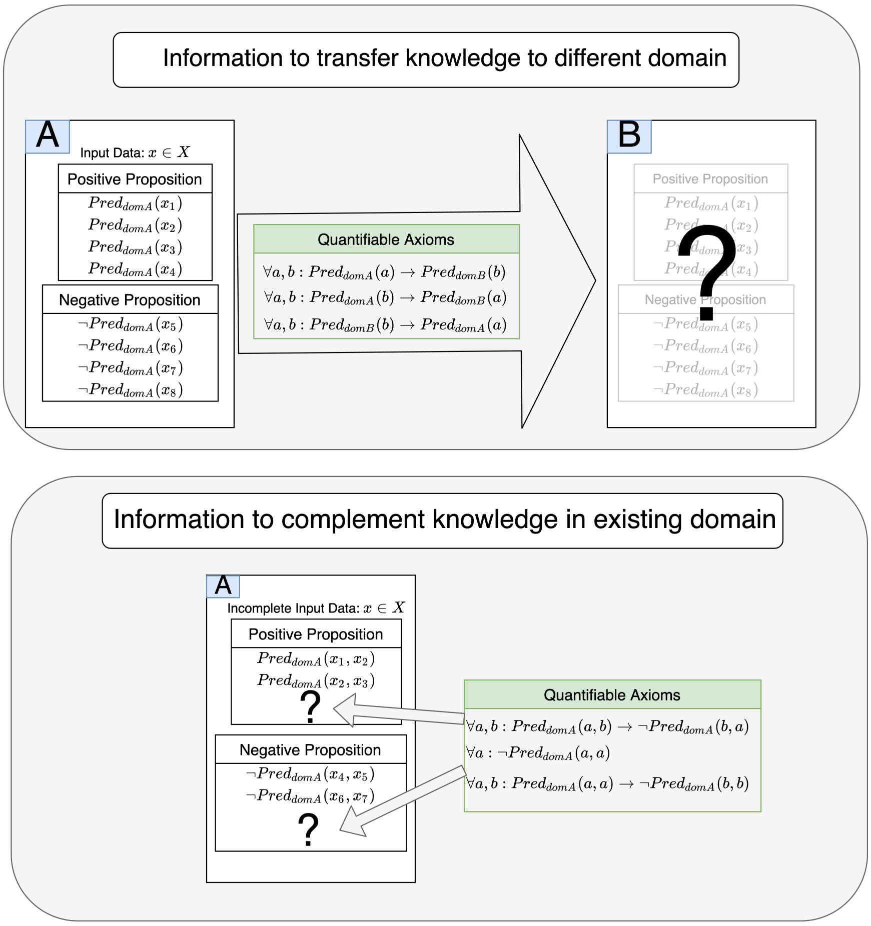

Information to transfer knowledge to different domain
A
Input Data: $x \in X$
Positive Proposition
$Pred_{domA}(x_1)$
$Pred_{domA}(x_2)$
$Pred_{domA}(x_3)$
$Pred_{domA}(x_4)$
Negative Proposition
$\neg Pred_{domA}(x_5)$
$\neg Pred_{domA}(x_6)$
$\neg Pred_{domA}(x_7)$
$\neg Pred_{domA}(x_8)$
Quantifiable Axioms
$\forall a, b : Pred_{domA}(a) \rightarrow Pred_{domB}(b)$
$\forall a, b : Pred_{domA}(b) \rightarrow Pred_{domB}(a)$
$\forall a, b : Pred_{domB}(b) \rightarrow Pred_{domA}(a)$
B
Positive Proposition
Negative Proposition
?
Information to complement knowledge in existing domain
A
Incomplete Input Data: $x \in X$
Positive Proposition
$Pred_{domA}(x_1, x_2)$
$Pred_{domA}(x_2, x_3)$
?
Negative Proposition
$\neg Pred_{domA}(x_4, x_5)$
$\neg Pred_{domA}(x_6, x_7)$
?
Quantifiable Axioms
$\forall a, b : Pred_{domA}(a, b) \rightarrow \neg Pred_{domA}(b, a)$
$\forall a : \neg Pred_{domA}(a, a)$
$\forall a, b : Pred_{domA}(a, a) \rightarrow \neg Pred_{domA}(b, b)$

Figure 6.3: Illustration of the different settings in which quantifiable axioms can complement existing data for a transfer into a new domain or to complement incomplete input data for the purpose of interpolation and extrapolation.

driven or knowledge-driven approach depending on the specific application domain. We propose an extension to the existing LTN framework where we are treating all logical formulas in the knowledge base equally to give the user the ability to assign preferences to each formula.

$$\mathrm{Sat}_A(\mathcal{G}_\theta(\mathcal{K}_w) = \frac{w_1 * \mathrm{Sat}_A(\mathcal{G}_\theta(\mathcal{K}_{prop}) + w_2 * \mathrm{Sat}_A(\mathcal{G}_\theta(\mathcal{K}_{quant})}{w_1 + w_2}$$

where $\mathcal{K}_{prop}$ entails all of the propositional knowledge and $\mathcal{K}_{quant}$ contains the general quantifiable clauses that enable transfer information (Figure 6.3). In this manner, the user is able to control the importance of specific information during optimisation, regardless of the number of propositional or general clauses that are provided. Similarly, further splits may be utilised so that negative propositions are equally weighted positives, irrespective of the sample size.

$$\begin{aligned}\mathrm{Sat}_A(\mathcal{G}_\theta(\mathcal{K}_w) = (&w_1 * \mathrm{Sat}_A(\mathcal{G}_\theta(\mathcal{K}_{proppos}) + w_3 * \mathrm{Sat}_A(\mathcal{G}_\theta(\mathcal{K}_{propneg}) \\ &+ w_2 * \mathrm{Sat}_A(\mathcal{G}_\theta(\mathcal{K}_{quantpos}) + w_4 * \mathrm{Sat}_A(\mathcal{G}_\theta(\mathcal{K}_{quantneg})) \\ &\div (w_1 + w_2 + w_3 + w_4)\end{aligned}$$

Thus, this weighting facilitates equal distribution of importance across different information, here differentiated into positive and negative propositional and quantifiable axioms.

The objective of this modular approach is not to create an additional hyperparameter w for each of the formulas in $\mathcal{K}$ as in Marra et al. (2020). Instead, we suggest a default weighting of equal importance to counter any imbalances. Essentially, the objective is to prevent special treatment of a particular type of information as doing so may lead to biased results. Because the relative importance of data, expressed as propositions, versus general knowledge, expressed as quantitative axioms, is highly dependent on the domain, we suggest equal treatment by default. Nevertheless, the ease of access to this parameter allows the user to customise the treatment of information accordingly.

In spite of the fact that the results of such optimisation may differ from the originally proposed LTN approach, where balancing of the information is not considered, the difference disappears for optimal solutions with sat-levels converging toward 1. As introduced by Serafini and Garcez (2016), the authors propose equal consideration for all entries in the knowledge base $\mathcal{K}$, which may lead to imbalances of relative importance for specific tasks. However, the updating of model parameters may still dynamically place relative importance on specific axioms through the chosen aggregation function. Regardless, a weighted grouping will be equivalent to the original LTN approach (Serafini and Garcez, 2016), when:

$$\mathrm{Sat}_A(\mathcal{G}_\theta(\mathcal{K}) = \mathrm{Sat}_A(\mathcal{G}_\theta(\mathcal{K}_w) = 1$$

Therefore, if both systems satisfy all of the axioms with a sat-level = 1, then their practical reasoning capabilities, which will be elaborated in the next section, will be equal. In practice, achieving a global maximum of sat-level = 1 may not be feasible with noisy samples of real-world data. Accordingly, prioritisation may be essential to incorporating the right reasoning capabilities into the system in situations where only a local optimum can be achieved.

Having provided a brief introduction into how different integration of knowledge through weighted clauses might be achieved and might lead to varying reasoning capabilities, we now wish to elaborate more on how precisely such reasoning capabilities may be defined and subsequently demonstrated through specific examples with both generated and existing datasets in comparison to other published work.

6.2.6 Practical Reasoning

Using the aforementioned method, the reader is well positioned to progress to sections 6.3 and 6.4 in which two practical application domains associated with XAI are discussed. Readers may choose to proceed to these sections if the real-world examples are didactically appealing. In this section, the method is supplemented by an examination of its reasoning abilities. We propose that in order to reason about what has been learned, it is worthwhile to discuss what the method can initially learn and deduce. We shall demonstrate how the system's ability to generalise to unseen cases and to deduce facts from learned observations depends on how logic formulas are formulated. Thus, we believe that establishing these limitations will provide a more robust foundation for our proposed framework. However, these subsections are not essential to fully comprehend the practical applications that follow in the sections on fairness (6.3) and concept explanations (6.4).

The term *reasoning* has been used frequently in the AI and ML literature recently to denote various inference tasks. It seems desirable to specify the way that reasoning can be performed within an AI system. The most intuitive definition is that of a system capable of generating conclusions from a given knowledge-base. In a seminal paper about reasoning in ML (Bottou, 2014), Leon Bottou argues that "instead of trying to bridge the gap between machine learning and sophisticated all-purpose inference mechanisms, we can algebraically enrich the set of manipulations applicable to training systems, and build reasoning capabilities from the ground up." More importantly, Bottou shows that there is continuity between algebraically rich inference systems such as logical or probabilistic systems and simple manipulations such as the mere concatenation of learning models.

By translating symbolic knowledge into regular loss functions, DFL can make reasoning a part of learning as argued by Bottou. In this setting, one can analyse how knowledge and data follow specific rules of inference by observing in which way the loss term influences model learning while establishing a way to measure the reasoning capabilities that these optimisations enable. Alternatively, one can extract knowledge explicitly from the trained model and evaluate by proxy the reasoning capabilities of the model on the basis of such extracted knowledge. Accordingly, we will refer to this as *reasoning about what has been learned*.

In the first setting mentioned above above, one evaluates both the network's forward (inference) and backward pass (learning). In the second setting, the focus is exclusive on the forward pass. Furthermore, while the former is interested in specific examples (local reasoning), the latter takes the view of reasoning capabilities obtained through a set of examples (global reasoning). In this respect, while Krieken et al. (2020) analyses each fuzzy operator at a time w.r.t. their ability to translate rules of inference into the neural network (based on specific examples by analysing the partial derivatives of logical subformulas), in this chapter, we take a broad perspective and evaluate reasoning w.r.t. a set of axioms being trained in a network through a set of examples.

Traditionally, reasoning in LTN is the process of searching for a set of parameters and groundings that satisfy a given logical proposition (Badreddine et al., 2020). Since, as discussed, in this chapter we are concerned with evaluating the capabilities of a trained network, in what follows we define practical reasoning in LTN as akin to the task known as inference in Machine Learning. We therefore focus on reasoning given a single grounded theory $\langle \mathcal{K}, \mathcal{G}_{\boldsymbol{\theta}} \rangle$ under the usual assumption that some sound statistical evaluation, e.g. cross-validation or bootstrapping, has been applied to select the best available grounding (set of parameters).

Given a logical axiom Λ to be checked on a LTN, we generate symbolically a set of formulas $\{\phi\}_\Lambda$ such that $\Lambda \models \phi$, where $\models$ denotes logical consequence. We say that a LTN *proves* formula ϕ, written $\langle \mathcal{K}, \mathcal{G}_{\boldsymbol{\theta}} \rangle \vdash \phi$, if the satisfiability of ϕ given the best available grounding is greater than a predefined real number q ($0.5 \leq q < 1$), that is:

$$\langle \mathcal{K}, \mathcal{G}_{\boldsymbol{\theta}} \rangle \vdash \phi \ \textit{if} \ \mathcal{G}_\theta(\phi) > q.$$

We then check whether $\langle \mathcal{K}, \mathcal{G}_{\boldsymbol{\theta}} \rangle \vdash \phi$ for all $\phi \in \{\phi\}_\Lambda$, which gives us a measure of the neural network's reasoning capability. For example, to check whether a NN satisfies the axiom of Modus Ponens, we define $\Lambda = \{A, A \to B \models B\}$. Given atomic formulas $P(x)$ and $Q(x)$, suppose that $\langle \mathcal{K}, \mathcal{G}_{\boldsymbol{\theta}} \rangle \vdash P(a)$ and $\langle \mathcal{K}, \mathcal{G}_{\boldsymbol{\theta}} \rangle \vdash \forall x(P(x) \to Q(x))$. Thus, $Q(a)$ is added to the set of formulas to be checked for satisfiability in the NN, i.e. $Q(a) \in \{\phi\}_\Lambda$; $\{\phi\}_\Lambda$ may contain therefore all the instances that can be obtained given a set of axioms Λ. Examples of such axioms may include transitivity, monotonicity and symmetry of relations.

Given a set of examples $\boldsymbol{E} \subseteq \{\phi\}_\Lambda$, the relation $\vdash$ allows one to measure the reasoning capability of network $\langle \mathcal{K}, \mathcal{G}_{\boldsymbol{\theta}} \rangle$ w.r.t. Λ by measuring the level of satisfiability of all the cases in $\boldsymbol{E}$ in comparison with the satisfiability of all ϕ in $\{\phi\}_\Lambda$. Suppose for example that one wishes to evaluate in practice the reasoning capability of a given neural network that has been trained to recognise images with various bounding boxes in them. Suppose further that it is useful to check whether the network has learned that, whenever a bounding box is inside another bounding box, which in turn is inside another bounding box then the first bounding box must be inside that last bounding box. In other words, this property P of bounding boxes satisfies the transitivity relation, namely, $\phi = \forall X, Y, Z : (P(X,Y) \land P(Y,Z)) \to P(X,Z)$. A set of examples $\boldsymbol{E}$ may contain a number of bounding boxes $b_1, b_2, ...b_n$ satisfying instances of the above transitivity relation, such that given for example $P(b_1, b_2)$ and $P(b_2, b_3)$, one concludes that $P(b_1, b_3)$ holds. Checking the network's reasoning capability w.r.t. property P includes checking whether $P(b_1, b_3)$ is satisfied by the network (soundness) but also quantifying the size of $\boldsymbol{E}$ as a proportion of $\{\phi\}_\Lambda$ (as a measure of completeness), in this case all possible cases of transitivity.

Notice how practical reasoning takes into account the ability of the system to interpolate and extrapolate. In what follows, the LTN network will be queried systematically to obtain a measure of its reasoning capability. Our definition applies directly to the reasoning tasks evaluated in Bianchi and Hitzler (2019), where the reasoning capability of LTN is studied in the context of a set of experiments.
Here, we allow such experiments to be generated systematically based not only on Prolog semantics but on any proof theory, in particular with open-world semantics. As previously mentioned, querying using first-order logic (FOL) clauses in LTN is not only a post-hoc explanation in the traditional sense. We argue that querying should form an integral part of an iterative reasoning process allowing for incremental explanation and system improvement through the extraction of knowledge guided by reasoning capability and data (c.f. Figure 6.2).

Before we demonstrate how this process works through experiments, we wish to establish underlying concepts related to assumptions that may affect the system's capabilities during learning.

6.2.6.1 Open world and closed systems

The Open World Assumption is typically associated with (machine) learning systems since an integral part of learning is discovering new knowledge, therefore knowledge should not be assumed to be complete (as in the Closed World Assumption). Nevertheless, in supervised learning, models are typically restricted to inferring labels based on a finite set of labels within a dataset. The model thereby learns to associate any data with a predefined set of labels based on a closed set of labels. Hence, while the models can exist in an open world setting, since they create mappings for any inputs $x \in \mathbb{R}$, the supervised training of these models may be viewed as a realisation of the closed world assumption of the models. A possible method of circumventing this problem is to recognise *unknowns*, which is referred to as open set recognition, but it is typically neglected in supervised Machine Learning settings. Mundt et al. (2020) examine this phenomenon in supervised learning and factors it into a continuous learning framework. The underlying assumptions of the system's framing need to be taken into consideration in continual learning settings as well as in the neural-symbolic systems since it has a direct impact on the system's effectiveness. Mundt et al. (2020) propose that beyond the closed world scenario, models cannot be expected to be evaluated solely on held-out data obtained from the training distribution. As a result, in domains where inference data cannot always be assumed to come from the same distribution, Mundt et al. (2020) differentiates three types of possible inputs:

- *Knowns*: inputs originating from the same distribution as the training data. Predictions made by the model are accurate and certain.
- *Known unknowns*: unknown inputs to the model that cannot be determined with certainty. An explicit label can optionally be assigned to examples that do not belong to the set of known concepts for explicit training. Uncertainty in prediction may indicate that a model is aware of its limitations.
- *Unknown unknowns*: unseen instances that belong to unexplored, unknown data distributions or classes for which the prediction is generally overconfident and incorrect

Differentiation of this type is analogous to the notion of predicate grounding that underlies all Differentiable Fuzzy Logic methods. Here, we commonly learn the predicates mappings using sampled data based on *knowns*. Depending on the learned predicate mappings of the model, inputs may have uncertain affiliations (*known unkowns*). However, instances that do not fall within the sampled data distribution and *known* predicates may be incorrectly assigned to learned predicates with high certainty (*unknown unknowns*).

Consequently, a consideration of this distinction between open and closed domains and the completeness of predicates becomes an essential part of applying Neural-Symbolic systems as it influences the reasoning capabilities. We will proceed to demonstrate how different assumptions can be integrated and result in varying deductive capabilities. By applying LTN to data, we are able to demonstrate different reasoning capabilities depending on the assumptions made. Moreover, we examine the influence of differing degrees of knowledge (in)completeness on reasoning.
We will outline and demonstrate potential reasoning limitations in the following sections using synthetic exemplary data, and then explain how the iterative approach can help overcome these limitations by comparing it to previously published work by Bianchi and Hitzler (2019). To be able to reason about what has been learned, we must identify the possible limitations of the system's capabilities as they influence the type of information that can be extracted from and distilled into the system.

6.2.6.2 A dissection of $A \vee B$

The following is a list of exemplary scenarios that illustrate how various assumptions, data, and knowledge can affect the reasoning ability of differentiable fuzzy logic systems. We intend to demonstrate how the same neural-symbolic system will have varying reasoning capabilities due to learning predicate groundings based on a sampled data distribution. The purpose of this section is to examine the particular differences of specific setups of data, knowledge, and architecture by way of visual representations of various scenarios.

Figure 6.4 illustrates the task of learning $A \vee B$ based on sampled data and a neural network based on differentiable fuzzy logic. In this section, we will attempt to provide a high-level understanding of why specific data and knowledge selection may have a significant impact upon the system's capacity to generalise and deduce.
The scenarios will have different data and knowledge to be learned, and require specific activation functions. These characteristics will result in different extrapolation abilities. In addition, each scenario is suited to either open- or closed-world assumptions in terms and deductive capability, as demonstrated by a specific entailment that will be checked. The accompanying graphs are intended to illustrate the neural network's learned functions and highlight possible deficiencies that may result from the derived solution. Our analysis will be followed by a series of experiments in the next section where we will apply the conclusions to practical applications.
As a starting point, we will ground predicates in real data as part of logical rules, as this will influence the system's inference and reasoning capabilities. Figure 6.4 illustrates an exemplary data *grounding* task. We sample data $a \in A$, $b \in B$, and finally $c \notin A \vee B$ (which will be referred to as negative examples for $A \vee B$). By introducing A and B as unary predicates, we can learn mappings of data that belongs to these predicates.
The knowledge to be learned here includes not only the individual groundings of A and B based on the training set T but also the logical clause $A \vee B$. Due to the nature of LTN, we can not only extract the truth value of an input belonging to $A \vee B$, but also dissect the groundings with respect to the predicates A or B individually. The

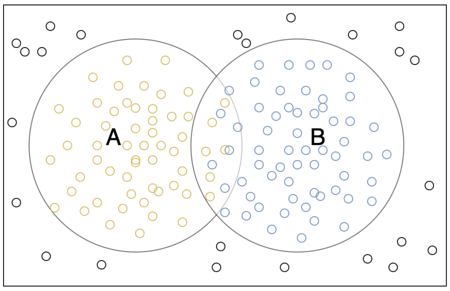


Figure 6.4: In our system, predicates are learned from data $a \in A$ (yellow),$b \in B$ (blue), and $c \notin A \vee B$ (black).

illustration shows the decision boundary of the predicate A in yellow and the decision boundary of B in blue.
We are explicitly interested in the behaviour of the intended entailment in each of these scenarios. Observing whether the deduced implication $\neg A \to B$ holds, allows us to assess our deductive reasoning abilities in different scenarios.
As a result of the different assumption, we need to differentiate between the query $\neg(A) \to B$ and the more precise query $(A \vee B) \wedge \neg A \to B$. While in a closed environment the domain of interest does not need to be specified, in an open environment it is required as will be demonstrated below.

Scenario 1

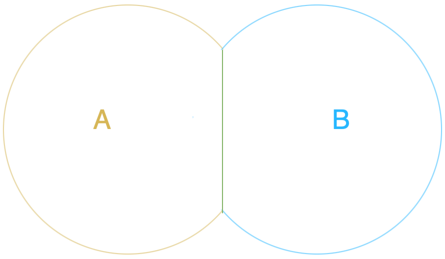


Figure 6.5: (Closed World): In this scenario, we are making the assumption that any data given must belong to either the domain A or domain B. This is due to the introduction of the *softmax* function at the output layer.

Knowledge Base: $\forall a, b : A(a) \vee B(b)$; no explicit negative examples

Output Activation Function: Softmax function

Extrapolation: All inputs will be assigned $A \vee B$. This means $\forall x : A(x) \vee B(x)$ will always be true for any real valued input

Assumption: Closed World

Entailment: $\neg A \to B$ always correct

Description: The assumption we are making in this scenario is that any data input will belong to either domain A or domain B. Training a model based on the singular axiom $\forall a, b : A(a) \vee B(b)$ may not yield meaningful results when confronted with out of distribution data as the model will confidently assume that the axiom is true at all times for every input. The precise query $(A \vee B) \wedge \neg A \rightarrow B$ is equivalent to $\neg A \rightarrow B$, which is always entailed within the underlying model architecture due to the *softmax* activation function at the output layer. The scenario presented here directly corresponds to the analysis of Mundt et al. (2020) that suggests the majority of classification tasks adhere to the closed-world assumption.

Scenario 2

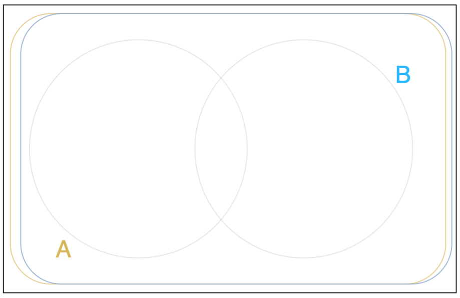


Figure 6.6: (Open World, without negative examples): This system assumes that the majority of the data belong to A and B. Although the entailment may have the desired property, it is a fallacy as $\neg A$ and $\neg B$ are mostly undefined since *Known Unknowns* are not taken into account.

Knowledge Base: $\forall a, b : A(a) \vee B(b)$; no explicit negative examples

Output Activation Function: Sigmoid function

Extrapolation: Undefined outside the samples a and b (training data distribution)

Assumption: Open World

Entailment: The precise query $(A \vee B) \wedge \neg A \rightarrow B$ should hold true although $\neg(A)$ may be unspecified for the input space leading to erroneous fuzzy truth values when evaluating on any real input

Description: The knowledge base consists exclusively of positive examples in this scenario. Accordingly, the model infers that A is always true and B is always true. Furthermore, $A \vee B$ will be true for all instances seen during training including the entailment $\neg(A) \rightarrow B$, although $\neg(A)$ is most likely not defined in this model grounding. It is important to note that this open-world system does not take into account any *Known Unknowns*. Outside of the training distribution, the model may incorrectly and confidently infer that instances belong to A and B.

Scenario 3

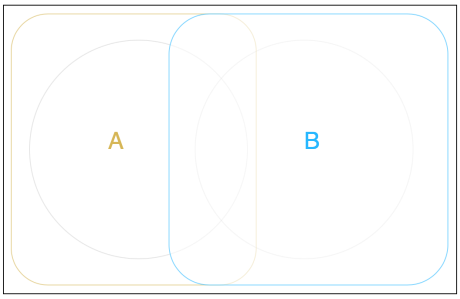


Figure 6.7: (Open World, with negative examples): By adding negative examples of A or negative examples of B, we are able to avoid that all data belong to both domains. Here we consider *Known Unkowns* which allows for better extrapolation as the model learns to differentiate between A and B.

Knowledge Base: $\forall a, b : A(a) \vee B(b)$; $\forall a : \neg B(a)$; $\forall b : \neg A(b)$; no data for $\neg A \wedge \neg B$

Output Activation Function: Sigmoid function

Extrapolation: Behaviour outside a and b is dependent on sample distribution for $b \notin A$ and $a \notin B$

Assumption: Open World

Entailment: $(A \vee B) \wedge \neg A \rightarrow B$ will hold true but only generalise if there are $b \notin A$ and $a \notin B$

Description: The addition of negative examples for A and negative examples for B allows the system to avoid that all inputs are assigned to to both domains A and B. Here, we consider a limited sample of *Knowns* as *Known Unkowns* for the individual predicates, allowing for more meaningful extrapolations as the model learns to distinguish between A and B. However, we do not consider any data that belongs outside of the sample distribution of A and B, which means that for $\forall x : \neg A(x) \wedge \neg B(x)$ the model may produce potentially undesirable results. The deduced entailment $\neg A \rightarrow B$ may be correctly recognised by the system provided that there are $b \notin A$ and $a \notin B$ in the sample distribution used for training.

Scenario 4

Knowledge Base: $\forall a, b : A(a) \vee B(b)$; $\forall a : \neg B(a)$; $\forall b : \neg A(b)$; $\forall c : \neg A(c)$; $\forall c : \neg B(c)$

Output Activation Function: Sigmoid function

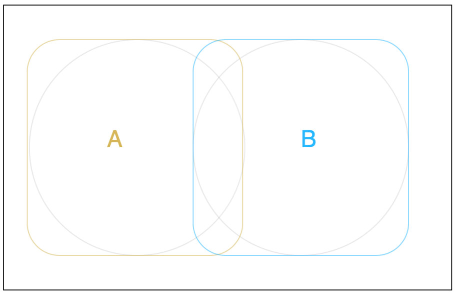


Figure 6.8: (Open World, with positive examples c): This scenario is comparable to scenario 3 but introduces *Known Unkowns* outside the predicates A and B of consideration. This should allow for better extrapolation and generalisation.

Extrapolation: Outside the *known* predicates A and B certainty of the model is dependent on the sample size of c where $\forall c : \neg A(c) \wedge \neg B(c)$

Assumption: Open World

Entailment: $(A \vee B) \wedge \neg A \rightarrow B$ will be correct. Precise query is needed in this open setting because $\neg A \wedge \neg B$ is in the data

Description: This scenario is comparable to scenario 3 but adds *Known Unkowns* outside of the previously considered predicates A and B to the knowledge-base. We include $c \notin A \vee B$ to explicitly learn about instances that are outside of $A \vee B$. Ideally, this should result in improved extrapolation and generalisation. If we want to query the deduced entailment, we need to be precise in our query as Differentiable Fuzzy Logic systems mainly use real data for quantification. In order to query the deduced entailment, it is important to be precise in the query as more data is sampled outside $A \vee B$.Thus, $\forall x : (A(x) \vee B(x)) \wedge \neg A(x) \rightarrow B(x)$ will ensure that we are only querying the behaviour within the domain of interest.

Scenario 5

Knowledge Base: $\forall a, b : A(a) \vee B(b)$; $\forall a : \neg B(a)$; $\forall b : \neg A(b)$; $\forall c : \neg A(c)$; $\forall c : \neg B(c)$; $\forall c : C(c)$

Output Activation Function: Sigmoid and softmax function

Extrapolation: Outside the *known* predicates A and B, everything becomes C, defining the unknown class (open set recognition)

Assumption: Open World

Entailment: $\neg A \rightarrow B$, unless $\neg A(x) \in C(x)$ (such as c). Correct entailment

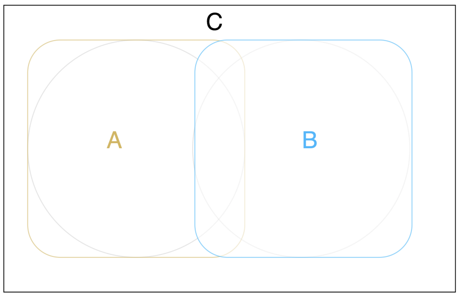


Figure 6.9: (Open World, with negative examples of A,B, and examples of C): This scenario is comparable to to scenario 4. However, we alter the architecture of the system in order to allow for improved extrapolation associated with open set recognition.

Description: The scenario described here is closely related to scenario 4. However, we modify the architecture of the system in order to incorporate enhanced extrapolation associated with open set recognition. This approach makes use of a *softmax* activation function to distinguish between *Knowns*, *Known Unkowns*, and *Unknowns*. As a result, the system is able to recognise the set it has learned about (A and B). The Softmax will ensure that any real data will be assigned either to the known class or to the unknown domains. After the system has determined that the data belongs to a known domain, a *sigmoid* is utilised to assign the inputs to a particular class within the known domain. As a result, the entailment $\forall x : (A(x) \vee B(x)) \wedge \neg A(x) \rightarrow B(x)$ will hold.

6.2.6.3 On the Raven's paradox

As a followup to giving a brief abstract overview of how different configurations of data, knowledge and architecture may lead to different capabilities, we would like to provide an additional example of how this effect may manifest in a different context as it pertains to different forms of reasoning.

Krieken et al. (2020) illustrate specific characteristics of reasoning capabilities in DFL systems with reference to the well-known raven paradox (Hempel, 1945). In order to approach the raven paradox, they specify and measure a class of differentiable updates which have particular properties. Specifically, the focus of the authors is on the fuzzy implications and adjustments to the relative importance of Modus Ponens (MP) versus Modus Tollens (MT) updates, which is central[2] to Raven's paradox. The paradox revolves around the question: What constitutes evidence for the statement that all ravens are black? According to first-order logic, $\forall x : raven(x) \rightarrow black(x)$.

In particular, the authors explore the problem associated with class imbalance in weakly supervised settings and distinguish between reasoning based on positive and negative (contrapositive) examples. The authors examine a number of gradient updates of differentiable fuzzy logic systems during the learning process and demonstrate

[2] Does the observation of non-black objects varied in colour and unrelated to ravens provide evidence that should increase the likelihood that all ravens are black?

that positive examples receive more prominence than negative examples during the updates. As a result, they argue that the majority DFL approaches may be less prone to contrapositive reasoning. In the context of the paradox with the above-mentioned background knowledge, this phenomenon can be illustrated when the model output is equivalent to a *non-black raven.* The distinction of MP and MT differentiable updates may be considered as follows:

- Modus Ponens: "It's a raven so it has to be black"; increase the truth-value of *black* during semi-supervised learning.
- Modus Tollens: "It is not black so it cannot be a raven"; decrease the truth-value of *raven* by contraposition of the implication.

The majority of fuzzy implications will be associated with MP reasoning (see first bullet point above). According to Krieken et al. (2020), this is undesirable, as MT reasoning (second bullet point) can be more representative of what occurs in practice in the real world.

While the debate regarding the importance of MT is bound to continue (MT does not appear in Prolog for example), the difference between MP and MT is particularly apparent in semi-supervised settings since the choice of Modus influences directly how unlabelled data is used for system optimisation. We argue that MT may need to be provided explicitly to the system in the form of a formula $\forall x : \neg black(x) \rightarrow \neg raven(x)$, allowing one to generate an adequate number of negative examples (non-ravens of various colours) to counteract the imbalanced gradient problem. When studying the Raven's paradox in the context of learning systems, non-raven examples are relevant and should be included as input to the system.

For optimal representational accuracy, direct access to the relevant data (instances of non-ravens, non-black things $\neg black(x) \wedge \neg raven(x)$) may be the preferred option to allow for a desirable interpolation, but the practicality of this may depend heavily on the domain. There may be limitations in an open world setting, as the non-black non-ravens may be difficult to specify as the domain may be harder to sample adequately. In spite of this, it should be noted that the observed phenomenon can be addressed both at the level of fuzzy logic and at that of data-driven analysis in the LTN iterative approach. It is evident that the problem can be greatly simplified when a closed-world assumption is adopted, and particularly when Clark's completion is assumed valid (not in the case of the raven's paradox). One can illustrate this with the well-known human reasoning example in which participants are told that *if there's an exam tomorrow then Lisa studies late in the library*, $\forall x : exam(x) \rightarrow library(Lisa)$. If participants are also informed that there will be no exams tomorrow, they conclude Lisa will not be in the library, which is a fallacy. A possible explanation is that most participants assume, when told *if there's an exam tomorrow then Lisa studies late in the library*, that the completion holds, i.e. *Lisa studies late in the library if and only if there's an exam tomorrow*, $\forall x : exam(x) \leftrightarrow library(Lisa)$. Given this closed-world assumption, it is not difficult to derive $\forall x : \neg exam(x) \rightarrow \neg library(Lisa)$, which provides an adequate knowledge-base for the generation of negative examples required for DFL learning.

In the following section, we will show how the proposed iterative LTN framework can be applied to address the imbalances and shortcomings discussed above. In contrast to the suggestion by Krieken et al. (2020) that appropriate operators are to be sought, we describe how a data-driven approach may provide an alternative means of solving the aforementioned challenges. Through a variety of experiments, we demonstrate that we can use the *continual querying* mechanism to learn about any flawed reasoning of the trained model and provide appropriate information to the system for correction. In this study, the reasoning capability of the neural network will be enhanced significantly, improving upon previous published benchmarks on Logic Tensor Networks Bianchi and Hitzler (2019).

6.3 Investigating the Neural-Symbolic Reasoning Capabilities

We now evaluate comparatively the reasoning capabilities of NNs in the LTN framework on the experiments used by Bianchi and Hitzler (2019) and Ebrahimi et al. (2021). In their experiments, redundant rules (formulas) are added to the knowledge-base. These are called redundant because they could be inferred symbolically from the existing knowledge-base. They conclude that redundancy produces an increase in the reasoning capability of NNs because the networks will see more data due to the additional rules. We demonstrate that only specific additional information is helpful in what concerns such improvement in reasoning.

In this section, the transition will be made to the practical applications of the proposed framework. Having discussed potential limitations of existing differentiable fuzzy logic methods, we now aim to demonstrate how the framework can be applied to address these shortcomings. Initially, this will revolve around logic-based problems, but will subsequently lead to the application of fairness in ML classification and computer vision. We intend to demonstrate the effectiveness of our approach over traditional XAI approaches in gaining a comprehensive understanding of the model behaviour as well as an interactive tool for guiding it.

In this section, we use the iterative continual framework of Figure 6.2 whereby a closed world or open world assumption can be implemented by adding specific propositions or quantified formulas to the knowledge-base. In a closed-world, it is assumed that information about a given domain is complete. The absence of information is therefore treated as negative information, or a license to *jump to conclusions* until further information to the contrary become available. By contrast, in an open-world, completeness of information is not required. By making an open or closed-world assumption with varying amounts of incomplete knowledge, reasoning capability results will vary. This may not seem surprising in the presence of the definition of practical reasoning we have introduced above, although to the best of our knowledge, this chapter is the first to address the issue systematically, which is important with all the recent interest in NN reasoning.

In LTN, a quantified formula either expands the set of examples in an existing domain (a form of semi-supervised learning) or it helps with transfer learning from a data-rich domain (defined by a set of predicates) to a domain with insufficient data (e.g. a new predicate). In the first experiment below, we illustrate the first case and in the second experiment information is transferred from a domain with high availability of data (parent) to one without (ancestor) as illustrated in Figure 6.3.

6.3.1 Ontology Reasoning

The first example illustrates the manner in which the reasoning capabilities of a system that presumes an open-world (NN) may be affected and assessed when it is applied in a closed-world setting. An ontology's *subclass* relations are given and quantified formulas are utilised to derive and complement data in order to infer multiple hop inferences on the ontology's knowledge graph.

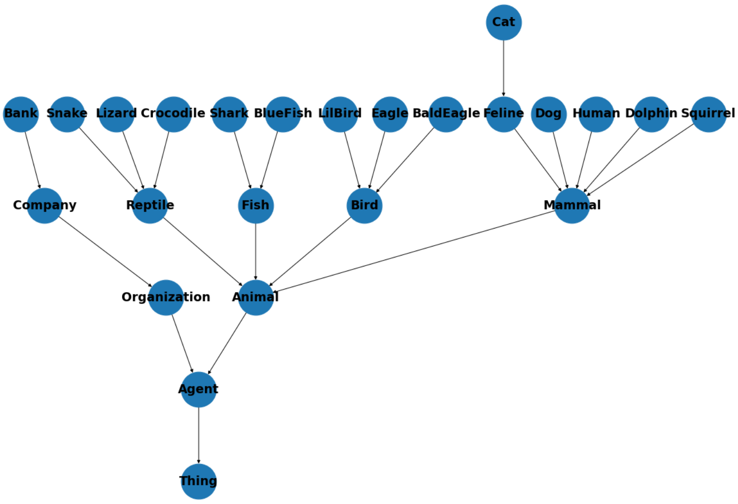


Figure 6.10: Exemplary taxonomy as proposed in Bianchi and Hitzler (2019) to learn how a NN trained using the LTN framework can perform on ontology reasoning. The task is to learn all subclass relations by extrapolating the subclass relation to any length.

Domain: classes, to denote the taxonomy entries.

Constants: $C_1, .., C_{24}$ denoting the 24 classes for which an embedding is to be learned.
$(C_1) = (C_2) = \cdots = (C_{24}) =$ classes.

Predicates: $sub(x, y)$ for the subclass relation. $sub \in$ classes.

Variables: x, y, z are variable ranging over the domain.
$(x) = (y) = (z) =$ classes.

	F1 score	Precision	Recall
Prolog	1	1	1
NN Revision 1	0.635	0.436	1
NN Revision 2	0.968	1	0.93

Table 6.3: F1 score, Precision and Recall of the deduced subclass relations; Prolog compared with LTN with two revision steps, the first without any negative examples and the second revision including negative examples.

Axioms Let $\mathcal{D} = \{(x, y)\}$ denote the 22 direct subclass relations, e.g. (Dog, Mammal), dog is a subclass of mammal. $\mathcal{D}$ is the training data for learning the the predicate groundings.

$$\forall x \in \mathcal{D} : \qquad \neg sub(x, x) \tag{6.5}$$

$$\forall x, y \in \mathcal{D} : \qquad sub(x, y) \rightarrow \neg sub(y, x) \tag{6.6}$$

$$\forall x, y, z \in \mathcal{D} : \qquad sub(x, y) \wedge sub(y, z) \rightarrow sub(x, z) \tag{6.7}$$

Let $\mathcal{E} = \{(x, y)\}$ denote the negative complement of the subclass relation. As Knowledge is assumed to be complete, it can be deduced by creating a set of all possible relations without the known subclass relation.

Grounding: $\mathcal{G}(\text{class}) = \mathbb{R}^2$. We learn embeddings in $\mathbb{R}^2$.
$\mathcal{G}(C_1 \mid \theta) = \mathbf{v}_\theta(C_1)$, $\mathcal{G}(C_2 \mid \theta) = \mathbf{v}_\theta(C_2)$, $\ldots$, $\mathcal{G}(C_{24} \mid \theta) = \mathbf{v}_\theta(C_{24})$; every class is associated with a vector of two real numbers. The embedding is initialised uniformly at random.
$\mathcal{G}(x \mid \theta) = \mathcal{G}(y \mid \theta) = \mathcal{G}(z \mid \theta) = \langle \mathbf{v}_\theta(C_1), \ldots, \mathbf{v}_\theta(C_{24}) \rangle$.
$\mathcal{G}(sub \mid \theta) : x, y \mapsto \texttt{sigmoid}(\texttt{MLP\_sub}_\theta(x, y))$, where $\texttt{MLP\_sub}_\theta$ has 1 output neuron.

Figure 6.11 illustrates the difficulty of integrating logical statements that originate from primary logic-based approaches directly into DFL systems without considering the open or closed world assumptions. Since there are only positive propositions in $\mathcal{D}$, the network grounds every variable positively to the predicate $sub(x, z)$ when applying formula (6.7). Training was done until there was no improvement in satisfiability for 50 subsequent epochs. Due to the fact that $\mathcal{D}$ only contains positive propositions, each variable in formula (6.7) is grounded positively to the predicate $sub(x, z)$ using the network. The formula does not provide any additional information for the gradient updates and, therefore, does not enhance the reasoning capabilities of the NN. Only formulas containing a negation in the consequent enable the network to acquire further relevant information in order to improve its reasoning capabilities. In the absence of such information, the network is unable to learn a meaningful interpretation of the vector embeddings. This is reflected in a low F1-score (F1=0.64) when considering all possible subclass relations (comparable to the results reported in Bianchi and Hitzler (2019)). Notably, the precision here is 0.436 but the recall is 1, indicating that the

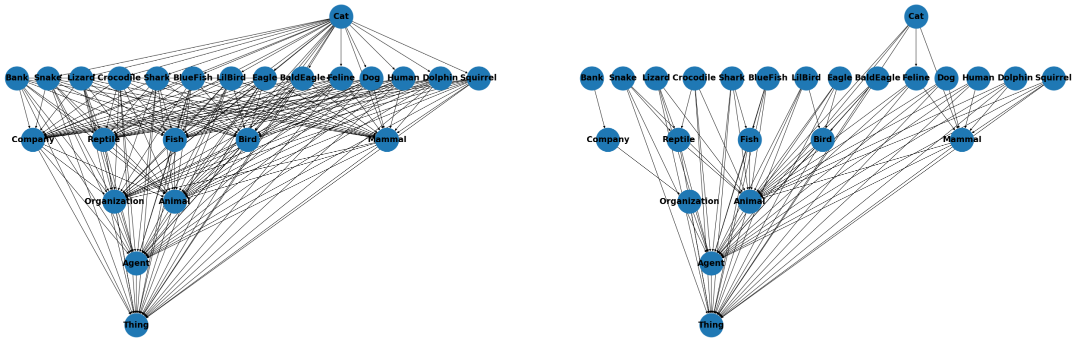


Figure 6.11: Exemplary taxonomy as proposed in Bianchi and Hitzler (2019) to learn how LTN can represent ontology reasoning and learn all subclass relations by extrapolating to any depth. The figure shows (on the left) subclass relations inferred before and (on the right) after negative propositions were added with F1 scores of 0.64 and 0.97, respectively.

network is prone to false positives. Nevertheless, by assuming that knowledge is complete, it is possible to deduce negative propositions $\mathcal{E}$ on the basis of transitive closure. By generating negative propositions for learning, we impose data based on a closed-world assumption upon an open-world model, leading to a significant improvement of the F1 measure (F1=0.968).

6.3.2 Reasoning in an Open-World

The goal of this task is to be able to use quantified formulas to complete an existing domain but also to transfer information effectively into a new predicate domain. Again, we compare results with Bianchi and Hitzler (2019), this time on the well-known ancestor example from Inductive Logic Programming. We further compare results with Prolog and the open-world theorem prover Coq, for the sake of a comparison with symbolic reasoning approaches. Using subsets of the knowledge-base $\mathcal{K}$ in Prolog and Coq, we can derive comparisons on reasoning capability with incomplete information and open domains. We shall also investigate continual learning within the LTN iterative approach and its ability to reason with partial knowledge. The iterative application of LTN allows one to increase the size of the knowledge-base incrementally by querying the network. The following example illustrates how false positives may be exacerbated over multiple deductive reasoning steps as part of this iterative approach.

The ancestor example is a well-known family tree problem centred around the learning of recursive relations, in particular the definition of *ancestor* as the recursive application of a *parent* predicate. Given only positive propositions about the parent relation, the ancestor relation is expected to be learned and to be generalised to the recursive application of any instance of the parent relation[3].

[3] In Prolog notation:

The task is a combination of transferring information from the parent predicate to the ancestor predicate while at the same time inferring the complement of the existing formulas (non-parents). This is achieved by the quantified formulas and it illustrates how false positives may be exacerbated over multiple reasoning steps.

Domain: people, to denote the individuals.

Constants: $C_1, .., C_{17}$ denoting the 17 individuals for which an embedding is to be learned.
$(C_1) = (C_2) = \cdots = (C_{17}) = \text{people}$.

Predicates: $par(x, y), anc(x, y)$ for the parent and ancestor binary relation, respectively.
$(par) = (anc) = \text{people}, \text{people}$.

Variables: x, y, z are variable ranging over the domain.
$(x) = (y) = (z) = \text{people}$.

Axioms Let $\mathcal{D} = \{(x, y)\}$ denote the 22 parental relations, e.g. (joe, juliet), joe is a parent of juliet, (janice, juliet), janice is a parent of juliet, etc. $\mathcal{D}$ denotes the training data. The knowledge-base is incrementally expanded using the 10 quantifiable clauses listed in Figure 6.12 (left) and below.

$$\begin{aligned}
\forall x \in \mathcal{D} &: & \neg par(x, x) \\
\forall x \in \mathcal{D} &: & \neg anc(x, x) \\
\forall x, y \in \mathcal{D} &: & par(x, y) \rightarrow anc(x, y) \\
\forall x, y \in \mathcal{D} &: & par(x, y) \rightarrow \neg par(y, x) \\
\forall x, y \in \mathcal{D} &: & anc(x, y) \rightarrow \neg anc(x, y) \\
\forall x, y, z \in \mathcal{D} &: & anc(x, y) \wedge par(y, z) \rightarrow anc(x, z) \\
\forall x, y, z \in \mathcal{D} &: & par(x, y) \wedge par(y, z) \rightarrow anc(x, z) \\
\forall x, y, z \in \mathcal{D} &: & anc(x, y) \wedge anc(y, z) \rightarrow anc(x, z) \\
\forall w, x, y, z \in \mathcal{D} &: & par(w, x) \wedge par(y, x) \rightarrow \neg par(z, x) \\
\forall x, y, z \in \mathcal{D} &: & \neg par(x, y) \wedge par(y, z) \rightarrow \neg anc(x, z)
\end{aligned}$$

Grounding: $\mathcal{G}(\text{people}) = \mathbb{R}^2$. We learn embeddings in $\mathbb{R}^2$.
$\mathcal{G}(C_1 \mid \theta) = \mathbf{v}_\theta(C_1)$, $\mathcal{G}(C_2 \mid \theta) = \mathbf{v}_\theta(C_2)$, $\ldots$, $\mathcal{G}(C_{17} \mid \theta) = \mathbf{v}_\theta(C_{17})$; every individual is associated with a vector of two real numbers. The embedding is initialized uniformly at random.
$\mathcal{G}(x \mid \theta) = \mathcal{G}(y \mid \theta) = \mathcal{G}(z \mid \theta) = \langle \mathbf{v}_\theta(C_1), \ldots, \mathbf{v}_\theta(C_{17}) \rangle$.
$\mathcal{G}(parent \mid \theta) : x, y \mapsto \mathtt{sigmoid}(\mathtt{MLP\_parent}_\theta(x, y))$, where $\mathtt{MLP\_parent}_\theta$ has 1 output neuron.
$\mathcal{G}(ancestor \mid \theta) : x, y \mapsto \mathtt{sigmoid}(\mathtt{MLP\_ancestor}_\theta(x, y))$, where $\mathtt{MLP\_ancestor}_\theta$ has 1 output neuron.

	Known True	Known False	Unknowns	Total
Revision 1	22	17	250	289
Revision 2	22	17	250	289
Revision 3	22	33	234	289
Revision 4	46	57	186	289

Table 6.4: Using the theorem prover Coq, we can derive proofs for true and false ancestor relation given axioms and propositions. It shows that although the information is incomplete, each revision step adds to what can be deduced.

As shown in Figure 6.12, the full knowledge-base is split into partial knowledge-bases and queried incrementally on knowledge that does not form part of the training process. The formulas used here correspond to those in the *small* and *extended* knowledge-bases used in Bianchi and Hitzler (2019). For efficiency, we use an embedding of size 2 as opposed to 10 in Bianchi and Hitzler (2019). Furthermore, we employ the product Real Logic operators introduced in Krieken et al. (2020). With the theorem prover Coq, we can derive proofs of the ancestor relation given a knowledge-base and propositions (i.e., examples). Coq derives these proofs under an open-world assumption. The information is incomplete, although each revision step contributes to the knowledge-base (Table 6.4).

	F1 score	Precision	Recall	Sat Level	Coq True Positives	Coq True Negatives	Coq Unknowns	Positive Unknowns	Negative Unknowns
LTN Rev. 1	0.597	0.426	1	1	22	17	250 (86.5%)	9.6%	90.4%
LTN Rev. 2	0.631	0.46	1	0.907	22	17	250 (86.5%)	9.6%	90.4%
LTN Rev. 3	0.875	0.84	0.913	0.912	22	33	234 (81%)	10.26%	89.74
LTN Rev. 4	0.884	0.84	0.933	0.945	46	57	186 (74.4%)	0%	100%
Prolog Rev. 1,2,3	0.647	1	0.47						
Prolog Rev. 4	1	1	1						

Table 6.5: F1 score, Precision and Recall of the deduced ancestor relations using Prolog and LTN for the revision steps depicted in Figure 6.12. Prolog achieves perfect scores as expected with all the required information provided under the closed-world assumption. Coq is unable to prove a large number of cases under open-world assumption. LTN is able to perform well under open-world assumption given incomplete information, while improving on the reasoning capabilities reported in Bianchi and Hitzler (2019) with each revision step of the iterative LTN.

In a first revision step, we initialise the task with minimal background knowledge consisting of the 4 formulas highlighted in the first column of Figure 6.12 and the 22 propositions ($\mathcal{D}$) of the parent relation. The network is trained at each revision step to learn a grounding for the clauses indicated by a green arrow in the figure. Upon training, we observe a fast convergence to 99% satisfiability of $\mathcal{K}$ after 2,000 epochs, whilst in Bianchi and Hitzler (2019) networks were optimised up to 10,000 and 20,000 epochs to converge.

By querying each clause after learning from clauses 1 to 4, it is evident that the

ancestor(X,Y) :- parent(X,Y).
ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y).

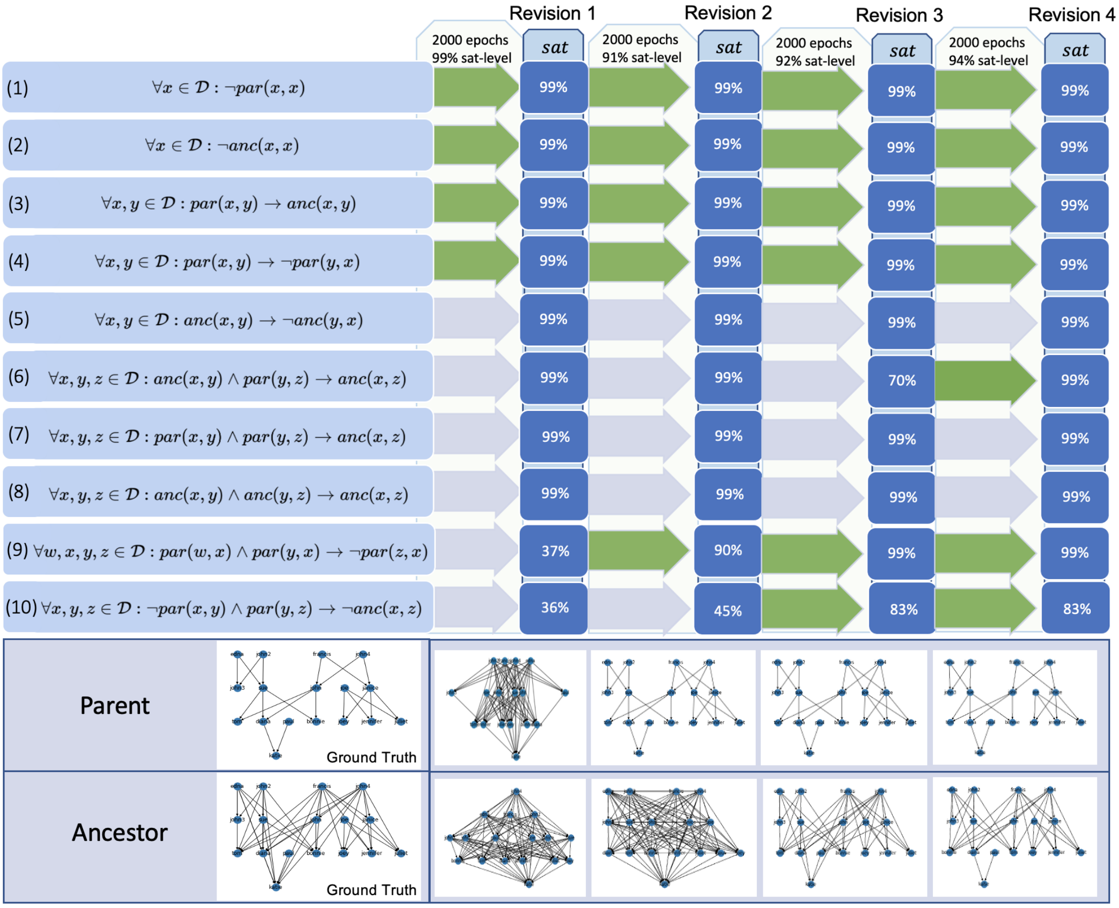

Revision 1
Revision 2
Revision 3
Revision 4
2000 epochs 99% sat-level
2000 epochs 91% sat-level
2000 epochs 92% sat-level
2000 epochs 94% sat-level
sat
(1) $\forall x \in \mathcal{D} : \neg par(x, x)$ 99% 99% 99% 99%
(2) $\forall x \in \mathcal{D} : \neg anc(x, x)$ 99% 99% 99% 99%
(3) $\forall x, y \in \mathcal{D} : par(x, y) \to anc(x, y)$ 99% 99% 99% 99%
(4) $\forall x, y \in \mathcal{D} : par(x, y) \to \neg par(y, x)$ 99% 99% 99% 99%
(5) $\forall x, y \in \mathcal{D} : anc(x, y) \to \neg anc(y, x)$ 99% 99% 99% 99%
(6) $\forall x, y, z \in \mathcal{D} : anc(x, y) \land par(y, z) \to anc(x, z)$ 99% 99% 70% 99%
(7) $\forall x, y, z \in \mathcal{D} : par(x, y) \land par(y, z) \to anc(x, z)$ 99% 99% 99% 99%
(8) $\forall x, y, z \in \mathcal{D} : anc(x, y) \land anc(y, z) \to anc(x, z)$ 99% 99% 99% 99%
(9) $\forall w, x, y, z \in \mathcal{D} : par(w, x) \land par(y, x) \to \neg par(z, x)$ 37% 90% 99% 99%
(10) $\forall x, y, z \in \mathcal{D} : \neg par(x, y) \land par(y, z) \to \neg anc(x, z)$ 36% 45% 83% 83%
Parent
Ground Truth
Ancestor
Ground Truth

Figure 6.12: Top: LTN solving the ancestor example Bianchi and Hitzler (2019) as part of an iterative revision process: $par(x, y)$ denotes that x is a parent of y; $anc(x, y)$ denotes that x is an ancestor of y. The figure shows the sat-level of each formula and the green arrows indicate the formulas used for training at each revision step. Bottom: Comparison of the graphs learned after each LTN revision with the ground truth of the ancestor example; the graphs approach the ground truth as more formulas that can generate negative propositions are retained by each revision.

network has acquired knowledge of clauses 5 to 8, as indicated by their satisfiability levels at revision 1 in Fig.6.12. Thus, using the extended knowledge-base as presented in Bianchi and Hitzler (2019) would not enhance the deductive reasoning capability in this case (with the use of the product real logic operators Krieken et al. (2020)). The network is generating many false positives (see Table 6.5) as numerous propositions and quantified formulas in $\mathcal{K}$ specify positive relations. The majority of unknown relations which cannot be deduced, as determined by the theorem prover Coq, are inferred to be true by the network. Although, under the assumption of a closed world, Prolog assigns all unknowns to false and achieves high precision (see Table 6.5). By querying clauses 9 and 10, whose consequents are negative literals, a low satisfiability value of around 37 % indicates also that the majority of parent relations that are not part of the training data are assumed to be true by the network. Therefore, either more data containing negative examples should be considered or specific quantified formulas should be added to LTN, both will yield the same effect of improving the groundings. Here we opted for the latter option due to the simplicity of adding a clause as a constraint to LTN. The incremental addition of clauses and the monitoring of performance improvements should make revision easier when clauses (rather than large chunks of data) may need to be removed later.

As part of revision step two, we add clause 9 having $\neg par(z, x)$ in the consequent to the LTN and continue to train the LTN for 2,000 additional epochs. Using this revised knowledge, new training examples will be created using the existing predicate and data. As shown in Figure 6.12 bottom, the added knowledge enables the network to eliminate false positives from the *par* relation. Notably, this translates into an improved inference of the *anc* relation by the network, as can be seen from the Table 6.5, despite Coq finding no additional proofs for these *anc* relation.

During revision step three, following an additional 2,000 epochs of training with the clauses shown in Figure 6.12, the $\neg anc$ relation is further specified. An examination of the F1 score in Table 6.5 indicates that the provision of negative examples significantly enhances the reasoning capabilities of DFL-models. Further specification of $\neg anc$ cannot improve Prolog's results where unknowns are assumed to be false under the closed-world assumption.

Lastly, revision four adds the clauses that fully specify all ancestor relations within the set of examples. All 46 positive *anc* relations can now be derived using the theorem prover Coq. However, the $\neg anc$ relations remain under-specified with 186 examples not being provable under an open-world assumption.

The continuous revision of the model has demonstrated that one can eliminate an undesirable model property by adding constraints that will increase the deductive capabilities reported in Bianchi and Hitzler (2019). Using less than 1% trainable parameters, we obtain a Mean Absolute Error of 0.04 and F1 of 0.884, while Bianchi and Hitzler (2019) achieved 0.14 and 0.85. The results indicate that open-world DFL methods are prone to recall while logical reasoners under a closed-world assumption are better at attaining high precision when the domain is underspecified. Considering this point is essential when making assumptions about the information required to ground a domain through learning. Our findings also indicate that DFL methods outperform logic-based methods when the information is incomplete.

Adding logically-redundant knowledge to a differentiable fuzzy logic system can be beneficial to improve overall performance. When the performance is dependent on or intended to measure the reasoning capability of the system, it is important to make the underlying assumption of a closed-world or open-world reasoning explicit. The first reasoning example used in this paper (ontology reasoning) showed how an open-world DFL model trained without sufficient negative information fails to learn simple relational knowledge. As expected, the iterative provision of negative information can improve reasoning capabilities dramatically.

The second example used in this paper (the ILP ancestor example) showed the same trend, with the iterative application of knowledge improving the reasoning capability of the neural network. The neurosymbolic approach showed also a superior reasoning performance than purely symbolic methods in the presence of incomplete information,

	F1 score	Precision	Recall	Sat Level
Prolog Rev. 1,2,3	0.647	1	0.47	
Prolog Rev. 4	1	1	1	
Coq (unk assumed false) Rev. 1,2	0.647	1	0.47	
Coq (unk assumed false) Rev. 3	0.647	1	0.47	
Coq (unk assumed false) Rev. 4	1	1	1	
Coq (unk assumed true) Rev. 1,2	0.289	0.169	1	
Coq (unk assumed true) Rev. 3	0.305	0.18	1	
Coq (unk assumed true) Rev. 4	0.33	0.197	1	
LTN Rev. 1	0.597	0.426	1	1
LTN Rev. 2	0.631	0.46	1	0.907
LTN Rev. 3	0.875	0.84	0.913	0.912
LTN Rev. 4	0.884	0.84	0.933	0.945

Table 6.6: F1 score, Precision and Recall of the deduced ancestor relation; Prolog compared with LTN for the revision steps depicted in Figure 6.12. Prolog is performing well as all required knowledge is given under closed-world-assumption. LTN is able to perform well under open-world assumption with incomplete information when compared with open-world Coq. Unknowns refer to all relations that are not provable under open-world assumption.

indicating how the combination of data and knowledge may benefit purely neural or purely symbolic (reasoning) approaches. Our results showed that the network was capable of capturing *multi-hop* steps and we intend to extend the evaluation to a large-scale knowledge graph with increasingly sophisticated relations.

Having demonstrated how the framework can enhance the reasoning capabilities of the AI system, we now show how the same concepts can be applied to the practical application domain of fairness in machine learning classification. The aim of the next section is to describe how revision and constraints can be used to remove biases from the classifier that are picked up from the training data. Further, we compare how this approach compares to existing fairness methods for neural networks and find that it outperforms comparative methods on three datasets while obtaining the same level of quantitative fairness.

6.4 Logic Tensor Networks for Fairness in AI

In this section, we propose an interactive neural-symbolic approach for fairness in AI based on the Logic Tensor Network framework. We show that the extraction of symbolic knowledge from LTN-based deep networks combined with fairness constraints offer a general method for instilling fairness into deep networks via continual learning. Explainable AI approaches which otherwise could identify but not fix fairness issues are shown to be enriched with an ability to improve fairness results. Experimental results on three real-world datasets used to predict income, credit risk in financial applications and recidivism show that our approach can satisfy fairness metrics while maintaining state-of-the-art classification performance.

More specifically:

- We introduce a method that allows one to act on information extracted by any XAI approach in order to prevent the learning model from learning unwanted behaviour or bias discovered by the XAI approach. We demonstrate how our method can leverage an existing XAI method, in this case SHAP Lundberg and Lee (2017), to discover and address undesired model behaviour. To recap: the goal of SHAP is to capture the average marginal contribution of a feature value across different possible combinations. A single Shapley value for a feature of this specific input denotes the contribution of such a feature to the prediction w.r.t. the average prediction for the dataset (Lundberg and Lee, 2017). The authors propose determining such a value by calculating the average change in the prediction by randomly adding features to the model. The Shapley value works for both classification and regression tasks, and is one of the most widely used XAI algorithms today.

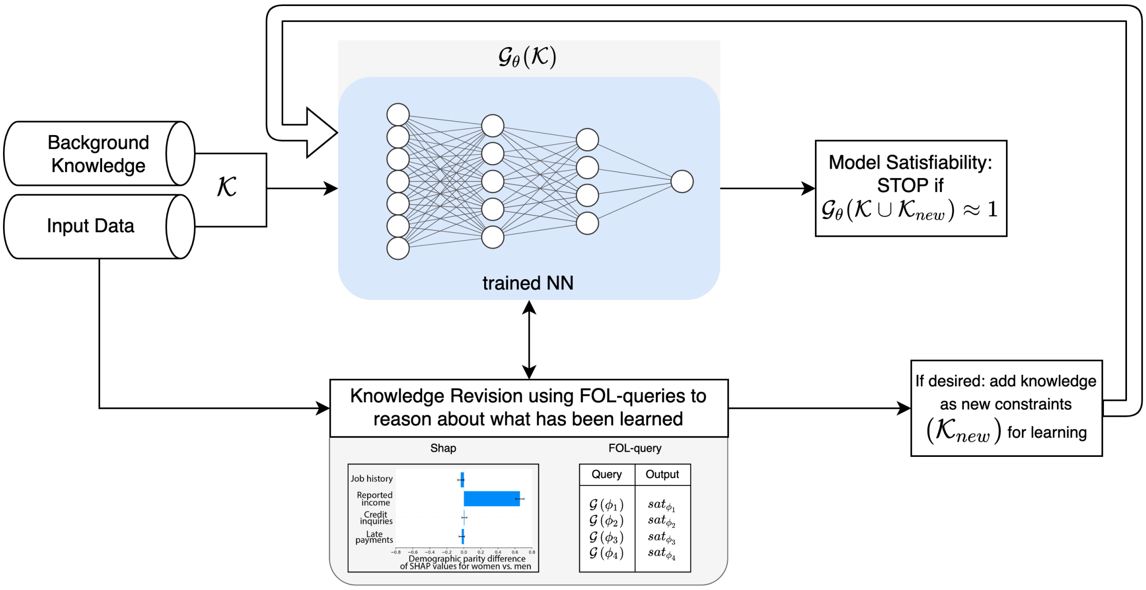
- We implement and outline the use of LTN for continual learning and iterative querying by caching the learned representations and by using network querying in first-order logic to check for knowledge learned by the deep neural network.

- We apply the proposed method and tool to the field of quantitative fairness in finance and recidivism in crime. Experimental results reported in this section show comparable or improved accuracy across three datasets while achieving fairness based on two fairness metrics, including a 7.1% average increase in accuracy in comparison with a state-of-the-art neural network-based approach Padala and Gujar (2020).

While the majority XAI methods make a noticeable contribution to the obstacle of gaining some understanding into model-behaviour, none of them address the problem of how one should act upon the extracted information. Consequently, we do not see the LTN approach as a method to be compared directly with the above XAI approaches but to complement them. By applying the neural-symbolic cycle multiple times, partial symbolic descriptions of the knowledge encoded in the deep network will be checked and, through a human-in-the-loop approach, incorporated into the cycle as a constraint on the learning process. This will enable an interactive integration of a desired behaviour, notably fairness constraints, by checking and incorporating knowledge at each cycle, instead of (global or local) XAI serving only to produce a one-off description of a static system.

6.4.1 Quantitative Fairness

One of the main goals of the recent advancements in explainability encompasses considerations of the fairness of automated classification systems. Although the discovery of such unwanted behaviour is essential and useful, in this section, we can address specific undesired properties, discovered or specified symbolically, and alter the learned model towards a fairer description.


$\mathcal{G}_\theta(\mathcal{K})$
Background Knowledge
Input Data
$\mathcal{K}$
trained NN
Model Satisfiability: STOP if $\mathcal{G}_\theta(\mathcal{K} \cup \mathcal{K}_{new}) \approx 1$
Knowledge Revision using FOL-queries to reason about what has been learned
If desired: add knowledge as new constraints $(\mathcal{K}_{new})$ for learning
Shap
Job history
Reported income
Credit inquiries
Late payments
−0.8 −0.6 −0.4 −0.2 0.0 0.2 0.4 0.6 0.8
Demographic parity difference of SHAP values for women vs. men
FOL-query

Query	Output
$\mathcal{G}(\phi_1)$	sat_{ϕ_1}
$\mathcal{G}(\phi_2)$	sat_{ϕ_2}
$\mathcal{G}(\phi_3)$	sat_{ϕ_3}
$\mathcal{G}(\phi_4)$	sat_{ϕ_4}

Figure 6.13: Illustration of the LTN pipeline for interactive continual learning: revision is carried out by querying a deep network interactively and learning continually, thus applying the neural-symbolic cycle multiple times. Explanations extracted from the network using e.g. SHAP can highlight bias in feature importance. Querying the network in LTN-style shows the satisfiability of fairness constraints which can be added to the knowledge-base $\mathcal{K}$ for further training. This process concludes once it has been shown to reduce bias highlighter through a subsequent SHAP explanation or the satisfiability of fairness constraints.

There have been a few methods addressing fair representation or classification: Dwork et al. (2012) seek to achieve fair representation in data instead of focusing on the classification. Agarwal et al. (2018) seek to achieve fair classification by proposing a reductionist approach that translates the problem onto a sequence of cost-sensitive classification tasks. Choi et al. (2019) study fairness in naive Bayes classifiers and propose an interactive method for discovering and eliminating discrimination patterns. Padala and Gujar (2020) integrate fairness into neural networks by including complex and non-decomposable loss functions into the optimisation. Fairness remains a significant challenge for Machine Learning. For an overview of the variety of fairness notions, we refer the reader to Zemel et al. (2013); Dwork et al. (2012). For an extensive overview of various fairness-oriented Machine Learning methods, we refer the reader to Friedler et al. (2019) and Mehrabi et al. (2019).

The above methods are related because they introduce constraints either on the data or the model during learning. The LTN-based approach used here introduces constraints as a regularisation which therefore may apply to any model or dataset. Also, in LTN, additional fairness axioms can be specified during training time by the user, which may be unrelated to the existing fairness axioms. Finally, at test time, the protected variables defined by such axioms are not used, so that a final customer of the ML system will not be asked for sensitive information on gender, race, etc.

Quantitative fairness metrics seek to introduce mathematical precision to the definition of fairness in machine learning. Nevertheless, fairness is rooted in ethical principles and context-dependent human value judgements. This functional dependence on value judgements is manifested in mutually incompatible definitions of fairness (Kleinberg et al., 2018; Lundberg and Lee, 2017). Rather than comparing extensively different notions of fairness, this section focuses on fairness as a current desiderata of explaining a models behaviour using XAI techniques, and therefore it uses the classical demographic parity metric and the legal notion of disparate impact to investigate the applicability of LTN.

The majority of fairness ML approaches can be considered to target group fairness, meaning parity between groups on aggregate. Dwork et al. (2012) advocate for the more fine-grained individual fairness where similar individuals should be treated similarly. Despite the focus on group fairness due to comparability, our approach can also be used to achieve individual fairness. We use the following standard definitions of group fairness to measure and compare with other methods. Following Agarwal et al. (2018); Padala and Gujar (2020), we consider a binary classification setting where the training examples consist of triples (X, A, Y) where $x \in X$ is a feature vector, $a \in A$ is a protected attribute, and $Y \in 0, 1$ is a label.

Definition 6.4.1. Demographic Parity (DP) A classifier h satisfies demographic parity under a distribution over (X, A, Y) if its predictions $h(X)$ are independent of the protected attribute A. That is, $\forall a \in \mathcal{A}$ and $p \in \{0, 1\}$

$$\mathbf{P}[h(X) = p \mid \mathcal{A} = a] = \mathbf{P}[h(X) = p]$$

Given that $p \in \{0, 1\}$, we can say:

$$\forall a : \mathbb{E}[h(X) \mid \mathcal{A} = a] = \mathbb{E}[h(X)]$$

the metric itself is typically reported as difference between both sides of the equation which converges towards 0 for fair classifier.

Definition 6.4.2. Disparate Impact (DI): Given (X, A, Y) as specified above, a classifier h has disparate impact if:

$$\frac{\mathbf{P}(h(x) > 0 \mid a = 0)}{\mathbf{P}(h(x) > 0 \mid a = 1)} \leq \tau = 0.8$$

when we use the industry standard “80%-rule” (Feldman et al., 2015). This metric compares the proportion of individuals from an unprivileged and privileged group that receive a positive output which converges towards 1 for full removal of DI between groups.

6.4.1.1 Further experimental descriptions

All experiments were run using the Tensorflow version 2.0 as well as PyTorch 1.2. The processor used i9 (4.5Ghz) with a NVIDIA 2080 TI GPU and 32GB Ram using Windows 10. Training is done until the satisfiability of the knowledgebase reaches 1 or the maximum of 3000 epochs is reached. Each of the experiments uses a regular feed-forward network with an output node for the predicate and two hidden layers of size 50 and 25 with a single output. The notebooks can be found in the accompanying repository inside the fairness subfolder.

As the datasets have been commonly studied across a variety of literature, we omit an extensive analysis of them Padala and Gujar (2020); Choi et al. (2019). Compass dataset has 6172 training examples after pre-processing (removing entries with empty data), Adult 45222, and German 1000.

All of the evaluation metrics used in our experiments were chosen to achieve comparability to the mentioned papers. We omit an extensive analysis of the appropriateness of the metrics at hand to solely focus on aligning with other methods. Regarding a discussions of the limitation of the fairness metrics, we refer the reader to (Dwork et al., 2012; Pleiss et al., 2017). Validation is done using 5-fold and 10-fold cross-validation to obtain comparability to (Padala and Gujar, 2020; Choi et al., 2019). In our experiments, choosing higher values for the aggregation than $p = 5$ can lead to NaN values in gradient back-propagation. This is a result of decreased precision when using tensor cores in the GPU of tf.float16. These errors do not occur when using CPU and can be avoided using lower values for p.

6.4.2 Experiment: Fairness using objective Features

The first experiment draws a parallel with a current state-of-the-art method in the area of explainability. We demonstrate how traditional XAI methods are able to benefit from a neural-symbolic approach. Most importantly, we show how the LTN method can remove any undesired disparities in a model-agnostic approach when having access to objective features as proposed in Dwork et al. (2012).

We use the same example as the authors of the popular SHAP library connecting XAI and fairness Lundberg and Lee (2017). For more detailed exposition of SHAP, we refer the reader to section 2.4.2. They aim to dissect the model's input features to understand the disparities based on quantitative fairness metrics in the context of a credit underwriting scenario.

A data generation process allows one to ensure that the labels are statistically independent of the protected class and any remaining disparities result from measurement, labelling or model errors. We generate four hypothetical causal factors scaled between [0-1] (income stability, income amount, spending restraint and consistency), which influence the observable features (job history, reported income, credit inquiries and late payments). The customer quality for securing credit is the product of all the factors that consequently determines the label as *high-customer quality* by being strong simultaneously in all factors. The observable features are subject to a bias introduced to obtain disparities in the system. This bias influences the mapping of the underlying factors to the observable features and therefore simulate an under-reporting of errors for women (the implementation contains further detailed explanation). We generate this synthetic data with 5000 data points for each gender.

We compare the demographic disparity between the gender groups by calculating their Shapley values. We use such values as a popular way of gaining insight into model behaviour, although other explainability methods could have been used, and show that one can intervene in the model by adding knowledge for further training of the LTN to reduce disparities. SHAP is used since it allows for detailed decomposition of feature-based disparities. However, any XAI method that is capable of highlighting undesirable disparities might have been chosen as an alternative method to motivate the addition of fairness axioms.

Since the SHAP method uses the same units as the original model output, we can decompose the model output using SHAP and calculate each feature's parity difference using their respective Shapley value. Then, by adding clauses to LTN which seek to enforce equality as a soft constraint, the neural network will be trained to reduce the difference (axioms 6.10-6.14 below). Re-applying SHAP would then give a measure of the success of the approach on parity difference.

Axioms 6.10-6.14 are created based on the idea of treating similar people similarly from Dwork et al. (2012). It is argued that finding an objective similarity or distance metric in practice can be challenging but should be possible[4].

First, we split the data into two subsets for the protected (F) and unprotected (M) group, respectively, and create five subsets within each group, denoted $\mathcal{R}_{Fi}$ and $\mathcal{R}_{Mi}$, $1 \leq i \leq 5$, using quantile-based discretisation of customer quality[5]. The five axioms (6.10 to 6.14) then state, according to the discretisation, that if a member (x) of set $\mathcal{R}_{Fi}$ defaults on credit, i.e. $h(x) = 1$, then a member (y) of set $\mathcal{R}_{Mi}$ should also default, $h(y) = 1$, and vice-versa. Due to our use of the LTN-based fuzzy logic, each satisfiability is determined by aggregation. If aggregation by average is employed (alternative aggregations are possible), then both protected and unprotected groups should default equally on average. Given the different groups, one may wish to specify that equality in prediction is required, e.g. for the bottom 20% of the protected group w.r.t. the unprotected group according to a fairness measure. In our approach, the use of the generalised p-mean lends itself very well to this task by allowing for different forms of aggregation for each equality sub-group (referred to as *customer quality* 1 to 5 below). As a result, the user can specify in the system how strictly each fairness axiom is expected to be satisfied (using the p-mean parameter p and the satisfiability threshold t, c.f. Algorithm 1). Experiment 1 is summarised below.

Predicate: D for the positive class (i.e. credit default)

Training data: $\mathcal{T}_D$, a set of individuals who credit default; $\mathcal{T}_N$, a set of individuals who do not credit default; $\mathcal{R}_{F1}, ..., \mathcal{R}_{F5} \subset \{\mathcal{T}_D \cup \mathcal{T}_N\}$, a set of female individ-

[4] The authors further advocate making such metric public to allow for transparency. They propose the use of a normative approach to fairness as the absolute guarantee of fairness.

[5] This choice should be attribute-independent and application specific, e.g. based on reported income from very low, low and medium, to high and very high, different groups may require different interventions, i.e. different axioms according to policy and the situation in the real-world

uals with customer quality 1 to 5; $\mathcal{R}_{M1}, ..., \mathcal{R}_{M5} \subset \{\mathcal{T}_D \cup \mathcal{T}_N\}$, a set of male individuals with the same customer quality.

Axioms:

$$\forall x \in \mathcal{T}_D : \quad D(x) \tag{6.8}$$

$$\forall x \in \mathcal{T}_N : \quad \neg D(x) \tag{6.9}$$

$$\forall x \in \mathcal{R}_{F1}, y \in \mathcal{R}_{M1} : \quad D(x) \leftrightarrow D(y) \tag{6.10}$$

$$\forall x \in \mathcal{R}_{F2}, y \in \mathcal{R}_{M2} : \quad D(x) \leftrightarrow D(y) \tag{6.11}$$

$$\forall x \in \mathcal{R}_{F3}, y \in \mathcal{R}_{M3} : \quad D(x) \leftrightarrow D(y) \tag{6.12}$$

$$\forall x \in \mathcal{R}_{F4}, y \in \mathcal{R}_{M4} : \quad D(x) \leftrightarrow D(y) \tag{6.13}$$

$$\forall x \in \mathcal{R}_{F5}, y \in \mathcal{R}_{M5} : \quad D(x) \leftrightarrow D(y) \tag{6.14}$$

Model: $h(x)$ (denoting $D(x)$) is the multilayer perceptron described in Section 6.2.3.

We initially train the multilayer perceptron on the data alone (without axioms 6.10 to 6.14) and observe the disparities shown in the SHAP XAI chart shown in 6.14 (left). The network learned undesired disparities among gender groups as a result of under-reporting errors in the data (equivalent to using only axioms 1 and 2 with an LTN trained for 1000 epochs).

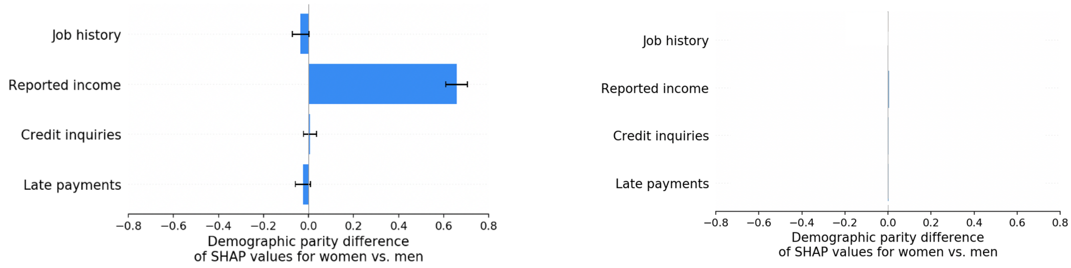


Figure 6.14: Disparity impact extracted with SHAP before and after LTN learning of fairness constraints.

We subsequently add axioms 6.10 to 6.14 to the knowledge-base and re-train the LTN with these axioms. As shown in the SHAP XAI chart of Figure 6.14 (right), this decreases disparity considerably, having reduced the Disparity Impact from 0.64 to less than 0.001. This illustrates the ability of LTN to account for fairness after having observed disparities in common XAI methods by adding appropriate fairness axioms for further training. Despite its usefulness as proof-of-concept, we acknowledge that the ideal notion of similarity among sub-groups can be impracticable. Next, we continue our investigation with additional real-world data and derive a notion of similarity automatically using the continual method.

6.4.3 Experiment: Fairness and Direct Comparisons on Real-World Data

We compare results on three publicly-available datasets used in the evaluation of fairness, obtained from the UCI machine learning repository: the *Adult* dataset for predicting income, *German* for credit risk, and *COMPASS* for recidivism. We follow the experimental setup used in Padala and Gujar (2020), although they perform extensive hyper-parameter tuning whilst our models are simpler. We compare our LTN-based approach with another neural network-based approach that integrates fairness constraints into the loss function using Lagrange multipliers Padala and Gujar (2020) (FNNC), and with an approach for naive-Bayes classifiers Choi et al. (2019). Gender is the protected variable in the Adult and German datasets, and race in the COMPASS dataset. We train a neural network with two hidden layers of 100 and 50 neurons, respectively (Padala and Gujar (2020) trains networks of up to 500 neurons per layer). We use the Adam optimiser with a learning rate of 0.001 trained for a maximum of 5000 epochs. As in Padala and Gujar (2020), we report results averaged over 5-fold cross-validation.

As before, we first train the network without fairness constraints, while before we relied on an objective notion of similarity that made it possible to split the individuals into sub-groups, now we use a continual learning approach (when such objective notion is not present). The trained network is queried to return the truth-value of the predicate used for the classification task $\mathcal{G}(D(\mathcal{T}))$ for the entire training set $\mathcal{T}$. The output helps determine, as a proxy for similarity, the fairness constraints. As done in the previous experiment, a quantile-based discretisation is carried out, but this time according to the result of querying the network, after splitting the data into two subsets for each class according to the protected variable. Therefore, we obtain equally-sized groups for each protected and unprotected variable. Again, five sub-groups are used and the axioms from Experiment 1 apply.

Querying axioms 3 to 7 reveals a low sat level at first as an indication of an unfair model ($sat_{\phi_n} < 0.5$). This is confirmed by measuring the fairness metrics with DI ≤ 0.4 and DP ≥ 0.03 across all datasets [6]. The results are shown in Figure 6.15 which also includes the results of the approach to account for fairness in naive Bayes classifiers Choi et al. (2019). The comparison with Choi et al. (2019) is not as straightforward as the comparison with FNNC Padala and Gujar (2020) because Choi et al. (2019) use 10-fold cross-validation and do not measure DI or DP. Nevertheless, we include the results of our approach using 10-fold cross-validation also in Figure 6.15 for the datasets. Since both LTN and FNNC are based on neural networks, we can make a more direct comparison with FNNC Padala and Gujar (2020).

As illustrated in Figure 6.15, we are able to outperform other state-of-the-art methods and achieve a lower variability across all datasets, and pass the DI and DP fairness thresholds proposed by Padala and Gujar (2020). All experiments were carried out using the same hyper-parameters as reported above and aggregation parameter $p = 5$.

[6] this is also revealed in SHAP value disparity. We measure the fairness here using pre-defined metrics as fairness is a known issue in these benchmarking datasets and therefore does not require an XAI method for detection

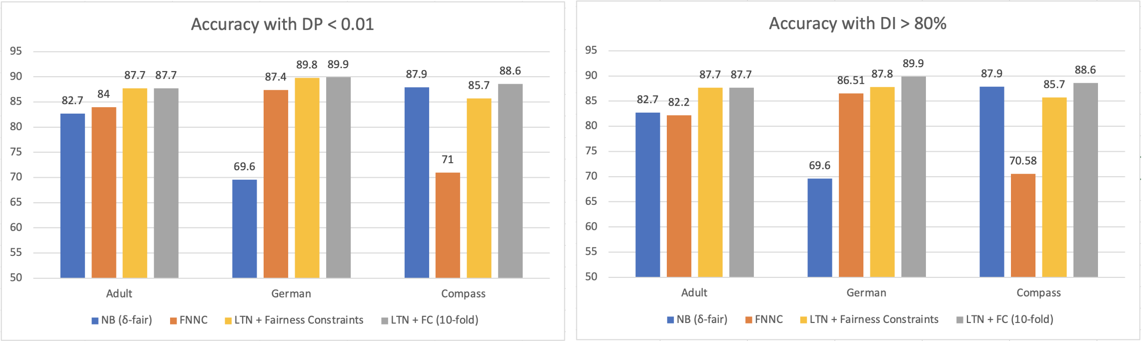


Figure 6.15: Comparative results of fairness-constrained learning: FNNC Padala and Gujar (2020), NB-based Choi et al. (2019) and the LTN-based approach proposed in this section using 5-fold cross-validation (LTN + Fairness Constraints or FC) and 10-fold cross-validation (LTN + FC 10-fold), on three datasets: Adult, German and Compass. The fairness metrics only apply to FNNC and LTN; NB-based uses a different metric for fairness and is therefore not directly comparable. The results of FNNC and NB are reported by the authors.

Finally, we would like to emphasize the flexibility of our approach w.r.t. different notions of fairness and its potential use with alternative fairness constraint constructions. The approach is not applicable exclusively to the metrics used here. With the increasing number and complexity of equality groups with larger p-values for aggregation, and the currently-evolving many notions of fairness being developed, we argue that rich languages such as FOL will be needed to capture more fine-grained notions, possibly converging towards individual fairness (with the generalised mean converging towards the *min* value).

Combining XAI methods with neural-symbolic approaches allows us to not only learn about the undesired behaviour of a model but also intervene to address discrepancies which is ultimately the goal of explainability in Artificial Intelligence. In this section, we proposed the interactive model-agnostic method and algorithm for fairness and have shown how one can remove demographic disparities from trained neural networks by using a continual learning LTN-based framework. While the first experiment demonstrated the effectiveness on addressing undesired gender-based disparities on simulated data, we have investigated such effectiveness on real-world data in the second experiment and compared to other methods.

Based on these results, it is clear that neural-symbolic integration can be useful for continual revision after reasoning about what has been learned. However, so far we have been limited to using the input values or the outputs of the model to specify the queries to the network and to establish constraints that influence the behaviour of the model. The user is thus severely limited in their ability to interact with the model. Hence, in the following section we will demonstrate how model inherent representation may be leveraged to arrive at more powerful logical descriptions of the inner workings of the system by utilising predicate groundings.

Chapter 7

Explainable AI using Concept Groundings

The experiments in the previous chapter highlighted how the querying may be used to understand the reasoning process of the neural network and therefore gain a better understanding of its inner operations. As opposed to existing methods, we demonstrate how a neural-symbolic approach to XAI may be used to incorporate knowledge back into a network to align the network's behaviour with the desired outcome and values. This section aims to bring all of the above together and integrate abstract concept grounding into the method in order to provide more intuitive understanding. We connect TCAV and LTN in order to allow for complex conceptual explanation and interactive learning. This enables domain experts to learn about the data-inferred decision making process of large ML models by querying the model and defining constraints for further learning as part of an iterative process.

In order to facilitate interaction and communication between AI systems and humans, we may draw inspiration from previous research on the use of language to communicate between cognitive agents (Cangelosi, 2006). Language in this context refers to a means for representing and communicating about the world. In Harnad (1990), the author draws attention to a problem known as the symbol grounding problem, in which natural and artificial cognitive agents must develop an intrinsic connection between their symbolic representations and some referents in the external world. In this process, individuals may represent external referents in a conceptual manner and use them as grounding for symbolic representations. The development of these internal representations occurs as a result of interactions with entities in the external world. Learning how to categorise allows us to form discrete and useful concepts of our environment. Based on the following chain of representations and entities, the cognitive mechanism underlying the grounding of physical symbols is formed (Cangelosi, 2006):

$$\text{external entities} \;\Leftrightarrow\; \text{internal representations} \;\Leftrightarrow\; \text{symbols.}$$

The relationship between external entities and representation and symbols is bidirectional, meaning that external entities affect representation and symbols, but symbols also affect how we represent. A practical translation of this can be accomplished by drawing an analogy from this.

$$\text{real world} \;\Leftrightarrow\; \text{neural network representation} \;\Leftrightarrow\; \text{concept groundings}$$

During learning, the most distinguishing characteristic of this type of perception is the requirement to warp the perceived into a similarity space of internal categorisations (Cangelosi, 2006). For the purposes of this process, we believe that a neural network is the most suitable method of learning transformations that may be capable of producing semantic representation spaces. Thus, we are interested in focusing on the bidirectional translation of neural networks into concepts. In the following sections, we illustrate how we approach this problem using the proposed framework.

In the following section, we further examine how we can use the LTN framework to reason about what has been learned by utilising inner representation. Our intent is to make use of concept groundings as symbols to integrate into the first-order logic framework of LTN. In this section, we wish to focus primarily on computer vision as the data domain since it may lend itself to a more intuitive framing of the proposed method as the notion of a concept in this domain is comparatively easily accessible. In spite of this, the technique is not limited to computer vision and we have shown, for example, how a concept might manifest itself in natural language.

Initially, we will be contextualising the different ways in which the framework may be used to provide comprehensive model understanding as well as the way in which the concept grounding fits into the framework and links to previous sections. This will be followed up with an investigation that will outline how robust representations of concepts can be established. Our goal is to arrive at transferable and generalizable representations as the basis for our conceptual groundings. We identify state-of-the-art models that enable such representations to be achieved.

After incorporating these into our framework, we demonstrate the ability of this method to provide a powerful explanation based on different datasets used in experiments. Additionally, the experiments will illustrate how a variety of models may be incorporated into the method as the method is not prescriptive with regard to the neural networks. Moreover, we demonstrate how concept representations can be used to influence a model's decision-making process. In this case, the concept representation enables straightforward communication with the model, since the user is able to specify how concepts should be applied to arrive at a decision.

This concept grounding is intended to provide accurate and accessible communication with the neural network. The section addresses many of the deficiencies discussed in chapter 3 & 4 and therefore provides a strong argument for using neural-symbolic communication to understandable Artificial Intelligence, where fuzzy logic provides the mathematical precision required to assess any formulated question or constraint in an accessible manner.

7.1 Practical Approaches to XAI using LTN

Following the outlining of the usefulness of Neural-Symbolic integration in mitigating the undesirable properties that may be identified by conventional XAI methods or the querying mechanism, we are now interested in highlighting how conceptual grounding may provide further advantages. Specifically, the purpose of this subsection is to contextualise the following proposition in light of the previous work of this thesis. It has been demonstrated above that LTN has the capability to provide post-training inferences for any query ϕ that allows for revisions to the system's reasoning capabilities. We have illustrated the benefit of performing such inferences while simultaneously training and building a knowledge base interactively. The following shall illustrate how the querying mechanism may be applied at different levels to achieve a variety of insights into the trained model. This will provide a high-level overview of how the proposed method that will follow complements the previously mentioned procedure for obtaining insights into the model.

As one approach, we may control a bounded variable x before aggregating $\forall x$ by restricting the selection of data to subsets that have distinctive features. In this sense, it is similar to the sampling methods introduced in the taxonomy. It has the advantage that LTN permits the use of FOL language which allows for simplicity and ease of use. The objective is to isolate the behaviour of a certain group in comparison to other groups. When combined with the fuzzy operators that enable fully differentiable Neural-Symbolic methods, this group-specific treatment is already an effective method for obtaining insights into model behaviour. The advantage of this approach is that it can be integrated over any model that acts as a grounding $\mathcal{G}$ in which the LTN represents the framework. In the fairness example, we utilise this mechanism to not only query the model behaviour of protected groups compared to unprotected groups, but also eliminate the undesired unfair treatment of specific subgroups using the targeted subgroups as aggregation filters.

We can formalise this notion as *targeted truth queries* where we intent to quantify over a set of elements of a domain for which the grounding satisfy some condition. Such a query consists of $(\phi_q(x), \mathcal{D})$, where x is a bounded variable of the formula ϕ_q and $\mathbf{D}$ are a set of examples where the dimensions of $\mathbf{D} = (\mathbf{d}^1, ..., \mathbf{d}^n)$ correspond to those of the domain of x.

There are two approaches to achieve an equivalent outcome. Either we can specify our *target* in our query ϕ_{q_t} as part of the formula as proposed in Badreddine et al. (2020) as *Guarded Quantifiers*. Essentially, the variables are dynamically masked according to a condition before the aggregation operators are applied.

We can translate the fairness example accordingly.

$$\forall x, y \in \mathcal{R} : D(x(risk = i)) \leftrightarrow D(y(risk = i))$$

For a more elaborate of explanation of the semantics of this type of quantification, we refer the reader to Badreddine et al. (2020). Our interactive LTN framework, including connectives and quantifier, follows the Real Logic as defined in Badreddine et al. (2020).

Alternatively, we propose the aggregation of specific closed variables if they satisfy a condition, which yields the same outcome in a static setting (applying the quantification mask on a fixed feature). Hereby, we specify $\mathbf{D}$ to consist of a set of examples where the condition holds $\mathbf{D_t} = (\mathbf{d_t}^1, ..., \mathbf{d_t}^n)$.

$$\forall x \in \mathcal{R}_{F1}, y \in \mathcal{R}_{M1} : D(x) \leftrightarrow D(y)$$

Among the fundamental limitations of such *targeted truth queries* is the constraint that we must specify the explanations for each distinct feature in the input space. The application of this method is unsuitable for complex tasks when the input data is very low-level, since a comprehensible explanation would require abstract conceptual representations.

Our proposal to address this is to extend LTN by incorporating conceptual representations that allow meaningful intuitive explanations. Essentially, the model is trying to assign concepts to learned distributed representations. By doing so, not only can intuitive explanations be generated comparable to human explanations, but also the procedures inside the model can be better understood. We are building on existing ideas, including the Concept-Activation-Vector presented earlier in the thesis as well as existing research on Linear Probes (Kim et al., 2018; Alain and Bengio, 2018). In our apporach, outputs and inner representations of concepts inside any neural network are mapped onto a logical predicate to obtain an explanation w.r.t. other predicates (other outputs and inner representations connected via logical connectives: negation, conjunction, disjunction, implication). Our aim is to query the neural network for symbolic knowledge so that a direct interpretation of abstract representations and operations on those representations become feasible.

In summary, if we consider the Differentiable Fuzzy Logic methods that use a subsymbolic model to ground low-level data into a symbolic representation, three distinct approaches to explanation can be derived:

- Interpretability on Predicate grounding level which is achieved through a combination of various model **outputs**. First-order logic can be used to formulate queries that connect predicate truth values (output). One illustrative example of this is the parent-ancestor relation above.

- Interpretability through targeted truth queries. This is done by closed aggregating on specific data. The aggregation of groundings is based on distinctive differences in **input** attributes. Illustrations of this approach are the fairness examples above. This allows us to study the effect, specific features have on the output groundings using bounded variables. In Badreddine et al. (2020), the authors propose an alternative for targeted querying using so-called *Guarded Quantification.* An example query would be $\forall y(\exists x : \text{risk}(x) = \text{risk}(y)(\text{G}(x) \leftrightarrow G(y)))$ The grounding is of the bi-implication are only aggregated for instances that satisfy the respective risk condition for x and y.

- Interpretability through **intermediate** model representations. By extracting inner embeddings, we can deduce how the model learned to perform a given task. It is possible to extract these representations from a neural network, for example,

> by localizing and translating their respective activation patterns. We can obtain a comprehensible description of the model reasoning process akin to human reasoning by integrating representations into conceptual groundings in First-Order-Logic. As will be demonstrated, the model classification of a zebra can be broken down into human-recognisable concepts where we dissect the process of recognising horses with stripes as zebras ($\forall x : horse(x) \wedge stripes(x) \implies zebra(x)$).

If we acknowledge that XAI is fundamentally concerned with establishing a form of communication between distributed representation based on various transformations and human-level reasoning, the main challenge becomes the translation between both levels.

The translation process can be divided into two distinct stages. In order to gather and make sense of low-level sensory data, conceptual entities have to be translated into symbols that refer to common concepts. An abstract concept is commonly used when a human is asked to explain why a perception task was performed by highlighting a distinguishing feature of an object. Furthermore, processes that provide information about concept interaction, such as the conjunction of concepts, need to be captured. It is through the logical operators that we gain an understanding of the type of relation concepts are in. Although an approximation, as it may not capture the complex interaction in its entirety.

7.2 Symbols and Conceptual Groundings

Symbols are tangible references which will be used to denote abstract concepts that arise through learning of model-specific, data-driven representations. These abstract concepts will be derived from within a trained model, giving rise to explanations that are grounded on the model's inherent representation and operations.

In fact, one of the core premises of the theory of communication is that communication is primarily symbolic (Littlejohn, 1977). According to symbolic interactionism, humans interpret and assign meaning to events through a set of symbols that are interconnected. Over the years, much research has been conducted on communication based on these principles, which can provide a broad framework for all forms of communication.Though it may seem desirable to communicate complex concepts in such an abstract way, the proposed approach in this chapter is more pragmatic and focused on obtaining general, reusable concepts. As an illustration of this, Figure 7.1 provides a variety of visual examples of the visual concept of a banana across different datasets. Once a model has learned a sound representation of a banana, it is capable of identifying this concept regardless of whether it is a photo, a sketch, an illustration, a ripe banana, or a banana cut into pieces. Even though these mappings are typically associated with network outputs, general representations of various concepts can also be identified within the network.

In recent years, Deep Learning research has focused on coming up with a transferable approach to Machine Learning, as catastrophic forgetting of acquired knowledge has proven inefficient and retraining deep model from scratch undesirable. As datasets

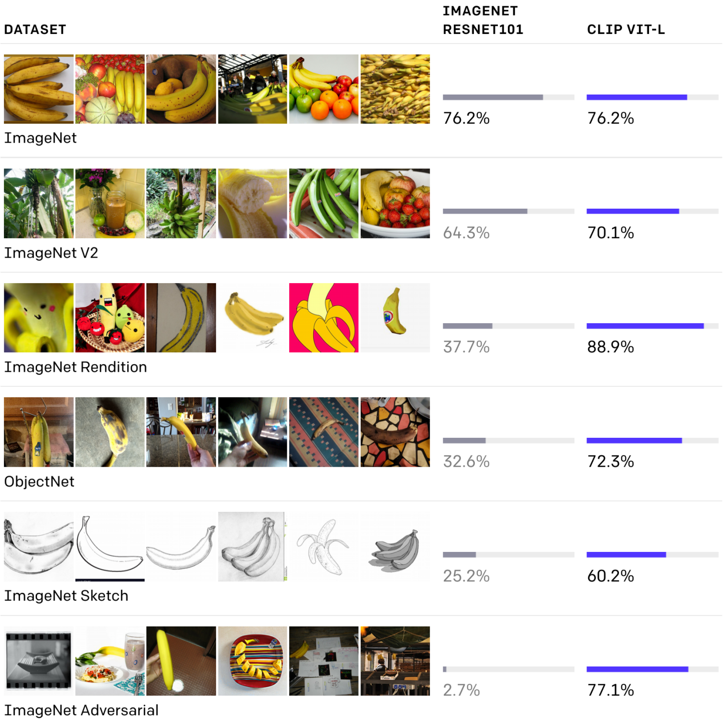


Figure 7.1: Illustration of the ability of CLIP to transfer learned information into a new domain (Radford et al., 2021). Here the distribution shift for bananas across different datasets is visualised. By demonstrating that the model can perform well across different domains demonstrates that a powerful representation is inherent to the model. For example, ImageNet Sketch and ImageNet Rendition are using abstract depictions of objects in the form of drawings, whereas ObjectNet produces objects in different angles using various backgrounds. The performance reported refers to the best zero-shop CLIP model, ViT-L/14@336px. The comparison model, ResNet-101, has the same performance on the well known ImageNet validation set.

and model sizes have exploded in recent years, it is becoming increasingly important to keep building upon learned information across various tasks. Particularly since the introduction of pre-trained convolutions and transformers, the emphasis has shifted from narrow end-to-end learning towards more general representation learning that can be applied to diverse downstream applications. In this setting, practitioners fine-tune models based on typically smaller datasets that had previously been optimized based on much larger datasets. With this approach, the aim is to learn a reusable representation, which results from a learned transformation that is transferable to a new domain.

It is possible to measure the degree of transferability of a model to a variety of tasks by looking at cross-dataset accuracy, since models that perform well across various tasks can be assumed to be based on general reusable concepts. In this section, we will discuss two types of models that have been particularly successful in transferring knowledge in such way.

It is well established that convolutional neural networks perform well in a wide range of computer vision tasks. In addition, studies have shown that certain components of a CNN are particularly useful for reusability across a range of applications Yosinski et al. (2014). A CNN may broadly be divided into two parts:

- Convolutional Component: usually composed of several convolutional and pooling layers that are stacked. The objective of this module is to generate and extract features from an image, thus, the output containing layers is referred to as a feature map (Zeiler and Fergus, 2014).
- Classifier: this part typically consists of fully connected layers. Based on the detected features, the classifier produces predictions. In a fully connected layer, every neuron has weighted connections to all of the outputs of the previous layer.

Initial studies were done to use pre-trained components of Neural Networks based on the intuition that a transformation must take place inside the network in order to map information from very general data (image with pixels) to specific classes (labels). Yosinski et al. (2014). Following this idea, the convolutional component extracts general features in the early layer, and more specific features in the later layer, where the classifier uses these feature mappings to make predictions. The neural network visualisation work of Zeiler and Fergus (2014) validates such claims. The common practice on the basis of this notion is to adapt pre-trained models to new tasks by replacing the classifier and repurposing the convolutional component in order to adapt to new tasks efficiently while incurring relatively low computational costs. This means that abstract concepts are integral parts of the CNN, and we shall demonstrate how general concepts map onto concepts in the following section.

Furthermore, the recent development of Transformer models offers yet another common practice of models built for the purpose of providing reusable representations. Originally popular in Natural Language Processing, this model architecture has since been adopted to the Computer Vision domain (Radford et al., 2021). Prior to Transformers, Machine Learning models were commonly trained using a specific dataset and components were reused, whereas in Transformers the learning of useful representations is the key component of the method. As shown in Figure 7.2, the representations acquired by these models have been shown to perform well on a wide range of datasets even without the requirement of further training (zero-shot).
Among transformers' major advantages is their ability to process enormous amounts of data. This is directly related to the optimisation procedure commonly used to train transformer models, including CLIP Radford et al. (2021). In contrast to models that are trained using datasets to assign single labels to images, CLIP is trained using natural-language image descriptions, as illustrated in Figure 7.3. Using the cosine similarity, both the image and text encoders seek to find a representation that matches each image and its corresponding description in text (using full sentences). The result is that the image is not only compressed into a single label, but also contains additional information, for example, multiple labels identifying each object in the image as well as the relationship between them. The ability to learn powerful

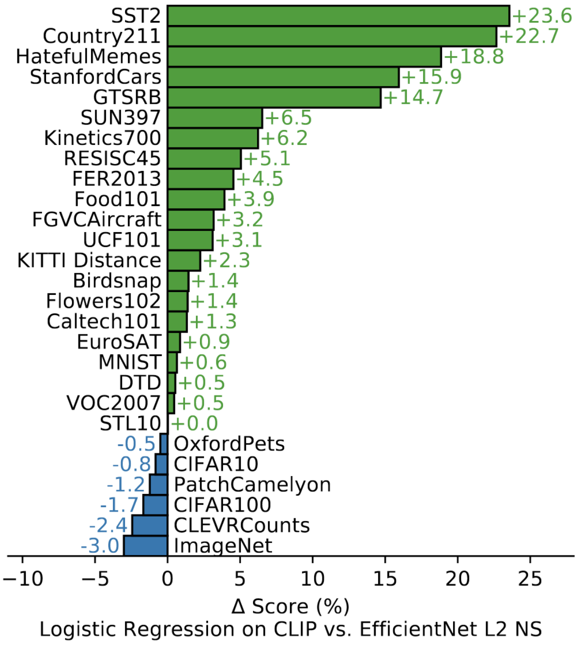


Figure 7.2: CLIP is capable of outperforming the current state-of-the-art ImageNet model using the rich feature representation. Classification is done using a linear layer on top of the feature output of CLIP. The comparison was done with the the Noisy StudentEfficientNet-L2 on 27 datasets by Radford et al. (2021).

mappings required for semantically rich representation is an outcome of the capacity to retain more information due to the architecture of self-attention. Furthermore, it requires less manual preparation of datasets for training since images with captions are more prevalent than images with single labels Khan et al. (2021).

In addition to being able to perform well across a number of tasks, a representation

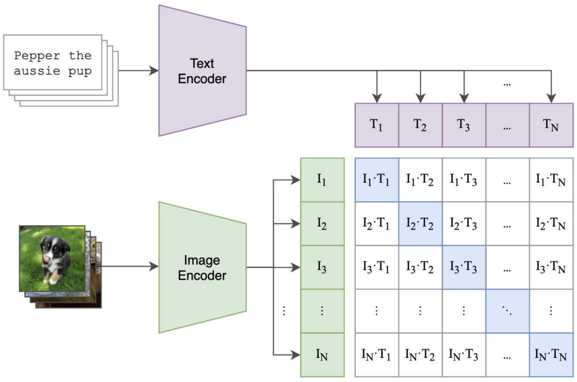


Figure 7.3: Illustration of the CLIP optimisation process (Radford et al., 2021). Images and their description are fed through their respective encoder. The dot product of encoded image and its encoded description is supposed to be maximised.

of this type is general enough to serve as the basis for an interactive and intuitive approach to explainability, as will be shown. By repurposing transformer and CNN models which are generally applicable across a wide range of datasets, we can uncover general concepts that can be used to form the symbolic representation of our explanations. In the next section, we will discuss how these concept representations can be integrated into our framework and used to develop an interactive intuitive explainability method.

7.3 LTN for Conceptual Explanations in Computer Vision

In the following, we will study three distinct methods that present strategies to derive an abstract representation from a Neural Network by utilising the model representations as outlined above. Each of these methods of representation will be incorporated into the LTN framework for interactive functionality, querying capabilities, and explainable extractions of information.

The first method will focus on convolutional neural networks and connect directly to the TCAV work by Kim et al. (2018). Fuzzy logic will be used to interconnect various concepts in order to address the shortcomings of chapter 3. The second method will demonstrate how to integrate an image encoder. It will establish that the method itself does not rely on a particular neural network architecture. Finally, we will outline how a multi-modal transformer can be integrated. The framework pipeline will be further customised to automatically arrive at common sense concepts by utilising external knowledge bases.

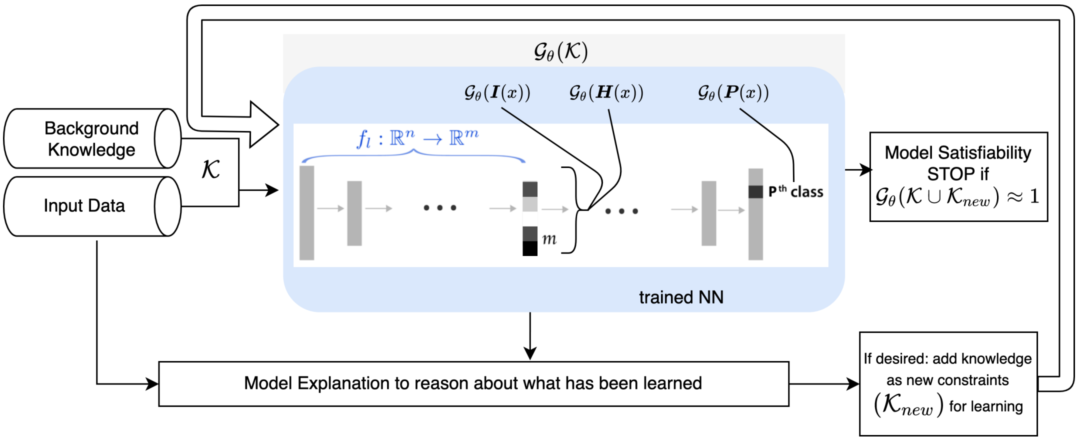
To obtain conceptual explanations that provide comprehensible descriptions of what has been learned, we must ground low-level information into reusable concepts that are present at hidden representations within the network. After outlining each approach briefly, we will subsequently demonstrate each of the proposed methods on real images and trained neural network in an experimental setting.

7.3.1 Concepts in Convolutional Neural Networks

In our initial approach which, we draw inspiration from the TCAV approach (Kim et al., 2018) but modify it substantially for the implementation in the LTN framework.

Consider any neural network that takes as input $\boldsymbol{x} \in \mathbb{R}^n$, which projects onto any layer l within the network consisting of m neurons, according to a function: $f_l : \mathbb{R}^n \rightarrow \mathbb{R}^m$. In an iterative explanation process, we seek to connect representations inside the network. There is no restriction on which layer l to use, but in a CNN this is generally the layer immediately before the fully-connected layer (i.e. the classifier) (Odense and d'Avila Garcez, 2020).

We adapt TCAV Kim et al. (2018) to enable users to specify the concepts to be checked (queried) at any adequate level of abstraction. Graziani et al. (2018) have shown in an application to medical imaging that domain-related concepts can be particularly valuable for gaining insight into the decision making process of models. The approach proposed in this section extends this idea to allow for complex concept



$\mathcal{G}_\theta(\mathcal{K})$
$\mathcal{G}_\theta(\boldsymbol{I}(x))$
$\mathcal{G}_\theta(\boldsymbol{H}(x))$
$\mathcal{G}_\theta(\boldsymbol{P}(x))$
Background Knowledge
Input Data
$\mathcal{K}$
$f_l : \mathbb{R}^n \rightarrow \mathbb{R}^m$
m
Pth class
trained NN
Model Satisfiability STOP if $\mathcal{G}_\theta(\mathcal{K} \cup \mathcal{K}_{new}) \approx 1$
If desired: add knowledge as new constraints $(\mathcal{K}_{new})$ for learning
Model Explanation to reason about what has been learned

Figure 7.4: The proposed interactive-learning LTN pipeline to provide explanations using concept groundings in hidden layer. We can use the feature map at any layer l to decompose the activation patterns into recognisable concepts. Subsequently the concept groundings may be put into relation with others and the output of interest using first-order logic. In this example we may use the grounding of abstract concepts I and H to explain class output P. Constraints can be added to train the NN continually until the reasoning process is in line with the user's expectation.

interactions (as defined by the logic) and model retraining using such logical rules as constraints. Using random examples alongside a user-defined set of examples (images) that capture a concept, we form a linear probe at layer l which evaluates the activation values produced by the examples and already known concepts from which further data can be selected for use.

In Kim et al. (2018), the linear probe at layer l serves as a building block for the Concept Activation Vector used to calculate conceptual sensitivities of inputs and classes. In this approach, we integrate the concept mapping directly into the interactive framework, allowing concepts to be combined into the logic, evaluated using fuzzy logic, and chosen for further training of the neural network model. Additionally, the mathematical properties resulting from fuzzy logic will be demonstrated to provide advantages over method-specific $TCAV$ scores.

The linear probe works as a classifier for our conceptual grounding, which is then integrated as a logical predicate into the LTN framework. Figure 7.4 illustrates the process. At each time that a user wishes to distil a model inference into specific concepts C, they simply need to select a set P_C of positive examples and a set N of negative examples. The linear probe then serves to distinguish the activation values of the neurons in layer l between $\{f_l(\boldsymbol{x}) : \boldsymbol{x} \in P_C\}$ and $\{f_l(\boldsymbol{y}) : \boldsymbol{y} \in N\}$.
This has the advantage of not being bound by pre-existing data or features. Independent of the original task, examples (images) may be collected and any number of user-defined concepts checked (queried) against the network. Inference is made based on activation patterns in the feature map of layer l that have been established during training.
Similarly to chapter 3, we construct an accuracy threshold that ensures that only

distinct activation patterns are grounded into the framework. We only consider concepts when the linear probe achieves high levels of accuracy above the threshold t, and can therefore distinguish positive from negative concept sets accordingly. As a result, concepts will be ignored when the underlying model has not learned a distinct concept representation. In this section, the concept accuracy threshold is set to if $t = 0.85$.

7.3.2 Concepts in Vision Transformers and Encoders

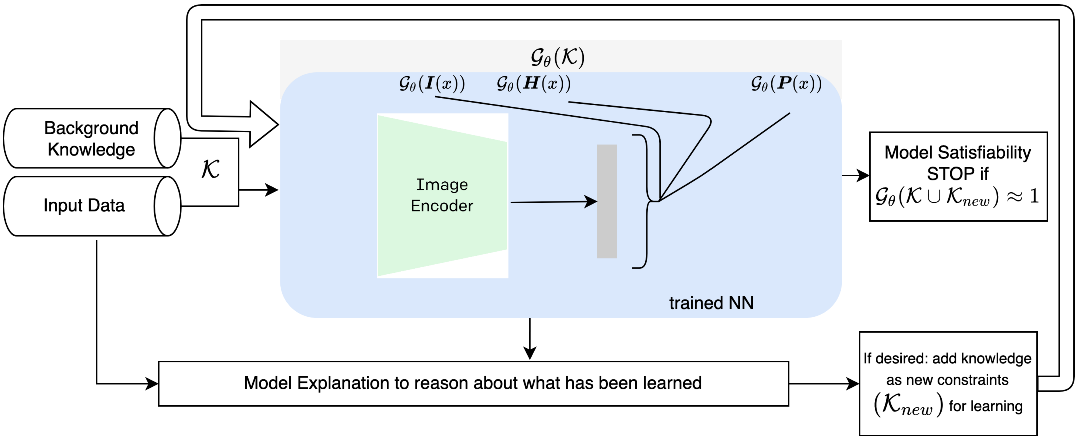

$\mathcal{G}_\theta(\mathcal{K})$
$\mathcal{G}_\theta(\boldsymbol{I}(x))$
$\mathcal{G}_\theta(\boldsymbol{H}(x))$
$\mathcal{G}_\theta(\boldsymbol{P}(x))$
Background Knowledge
Input Data
$\mathcal{K}$
Image Encoder
trained NN
Model Satisfiability STOP if $\mathcal{G}_\theta(\mathcal{K} \cup \mathcal{K}_{new}) \approx 1$
Model Explanation to reason about what has been learned
If desired: add knowledge as new constraints $(\mathcal{K}_{new})$ for learning

Figure 7.5: In this approach, we use the encoding of a vision transformer to ground the abstract concepts and classes for outputs. We train linear probes on the rich feature embedding of the CLIP model by Radford et al. (2021). As previously this allows us to explain the predictions using first-order logic formulas that contain abstract concepts.

Whereas in the previous approach feature mappings have to be extracted in an intermediate layer, the CLIP transformers and encoders can be integrated directly using the learned representations for concept groundings (see Figure 7.5).
As before, we employ a linear probe implemented as a single linear layer with sigmoid output in order to decode activation patterns generated by the image encoder and integrate them into the fuzzy logic framework. For the activation patterns to be extracted, a user-defined set P_C of positive examples is provided for each concept C, similar to the CNN extraction procedure. Figure 7.5 illustrates this integration, where $\mathcal{G}_\theta(\boldsymbol{I}(x))$ refers to the grounding of concept C_I and $\mathcal{G}_\theta(\boldsymbol{H}(x))$ to C_H.
Since the transformer and encoder were trained on a large number of images, the image encoder is capable of producing vector representations that contain a vast amount of information. Furthermore, the transformer learns a mapping C the image and a natural language description, and not just an individual label based on the standard 1-of-N label.
Learning from natural language has several advantages over other methods of training. Natural language as a training target increases scalability significantly over standard labelling image classification datasets, as Radford et al. (2021) can integrate

vast amounts of text-labeled images from the web. Additionally, the benefit of this method over most unsupervised or self-supervised methods is the flexibility of zero-shot transfer across domains, as not only a representation is learned but it is directly linked to natural language. As a result of this connection, a suitable integration for explainable concept grounding is possible. Natural language descriptions provide a good fit for the foundation of the conceptual explanations we seek in our proposed method.

Furthermore, the CLIP transformer also allows the direct encoding of text, meaning

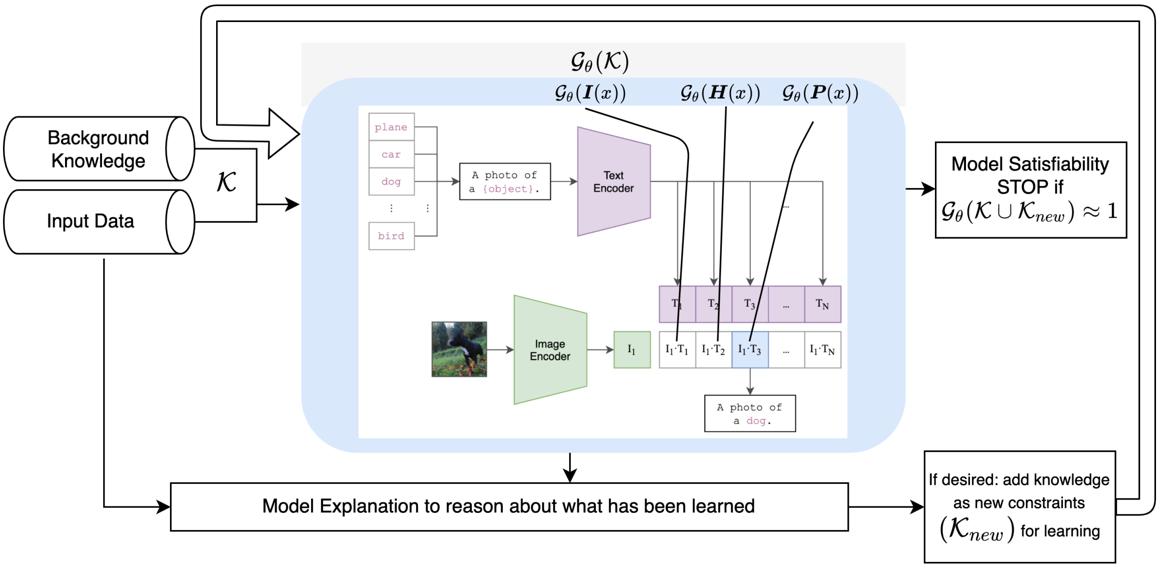

$\mathcal{G}_\theta(\mathcal{K})$
$\mathcal{G}_\theta(\boldsymbol{I}(x))$
$\mathcal{G}_\theta(\boldsymbol{H}(x))$
$\mathcal{G}_\theta(\boldsymbol{P}(x))$
Background Knowledge
Input Data
$\mathcal{K}$
plane
car
dog
bird
A photo of a {object}.
Text Encoder
Image Encoder
I_1
T_1 T_2 T_3 ... T_N
$I_1{\cdot}T_1$ $I_1{\cdot}T_2$ $I_1{\cdot}T_3$ ... $I_1{\cdot}T_N$
A photo of a dog.
Model Satisfiability STOP if $\mathcal{G}_\theta(\mathcal{K} \cup \mathcal{K}_{new}) \approx 1$
Model Explanation to reason about what has been learned
If desired: add knowledge as new constraints $(\mathcal{K}_{new})$ for learning

Figure 7.6: This approach allows for zero-shot grounding of concepts and learning of classes. We can use the text encoder in addition to the image encoder to measure cosine similarity of encoded image and description to find the inferred output.

that each concept in C may be directly computed. It will suffice in this case to use a natural language description of the concept for obtaining output values for the conceptual groundings, as shown in Figure 7.6. In the first step, CLIP will compute the image feature embedding and the text embeddings of all given text descriptions using the appropriate image and text encoders. The cosine similarity between the text and image embeddings is calculated and normalized using the temperature parameter τ in order to obtain a softmax probability distribution.

In the section on reasoning capabilities, it is outlined that the network needs "negative" examples to allow for sensible reasoning about the concepts learned under open-world assumption. This is accomplished by incorporating random or user-specific counter-examples in the form of a negative image- or text set N in the CNN explanation as well as the CLIP image encoder. Due to the zero-shot approach employed by CLIP for text and image integration, the counterexamples here are provided using alternative descriptions as text, as will be shown in the next section. Counterexamples can be provided manually or derived using external knowledge-bases.

As opposed to the original CLIP that focuses solely on learning representations based on scale, we provide logical explanations for image classification while using these representations. We will demonstrate how the conceptual dissection and integration using logical connectives enables the construction of a model that is truly explainable, accessible, and powerful.

7.4 Experimental Results

The experiments presented in this section demonstrate how the method can be applied to different neural network architectures. The four different integrated architectures should provide the reader with an indication of how this approach generalizes across various neural network designs. The integration into our framework consists of pointing predicate truth values to either the neural network output nodes or to a linear probe output. Class labels are usually outputs from a neural network, unless specified otherwise. Typically, model-inherent abstract concepts are specified using linear probes.

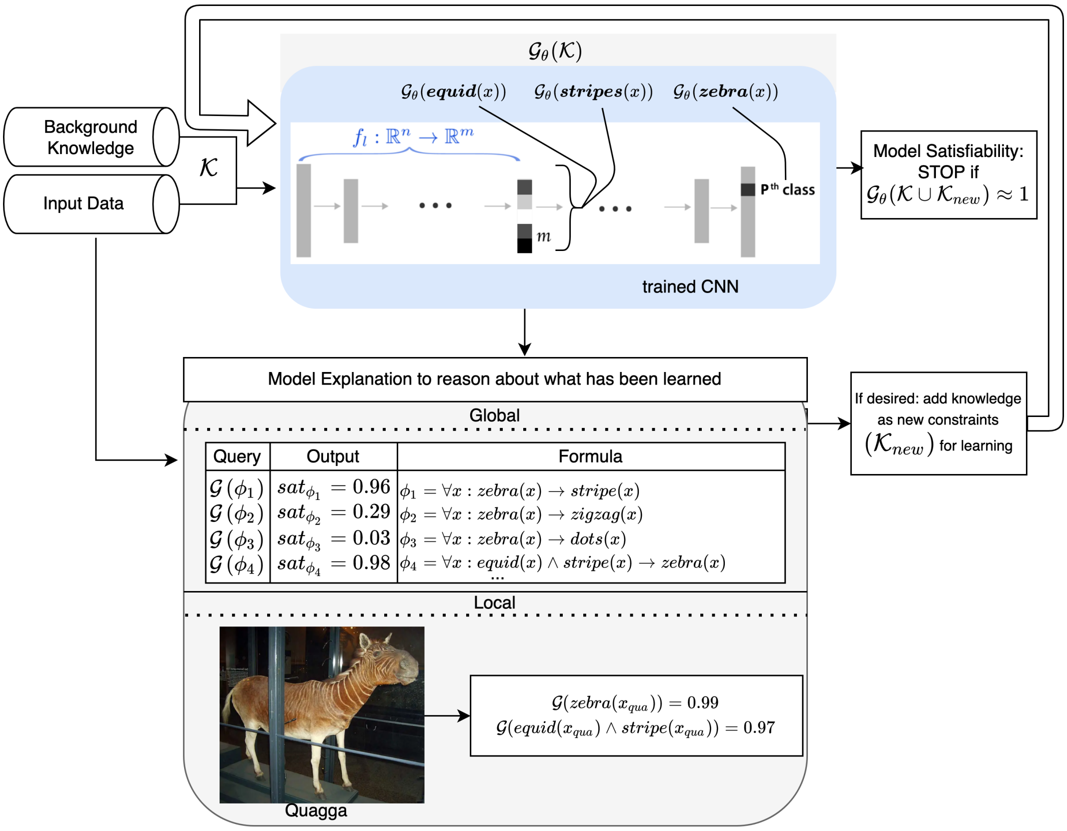
In each section, examples used to extract and test the concept representation will be outlined. When we used a pretrained network, we refer the reader to the original publication for more information regarding the data used for training. Evaluation of the knowledge base is performed using the following fuzzy logic operators: product operator, reichenbach norm for implication, and generalised mean with p=2 for aggregation. Training was done until there was no improvement in satisfiability for 5 subsequent epochs or a maximum of 50 epochs unless specified otherwise. The Adam optimiser is used with a learning rate of 0.01. The number of training examples for each concept will be reported in each section.

As we use pre-trained models, the outputs we explain have already been trained. Specifically, we train linear probes to distinguish between activation patterns. A separate test set is used to evaluate the subsequent queries explaining the relationship between concepts and outputs.

7.4.1 Convolutional Neural Network

In our first example, we establish a comparison directly with Kim et al. (2018) in order to demonstrate the effectiveness of the method we propose. In this example, we query a GoogLeNet model (Szegedy et al., 2015) trained on ImageNet to explain the output class of zebras with respect to user-defined concepts, as illustrated in Kim et al. (2018) and the bottom part of Figure 7.7.

Our initial objective is to compare the explanation on the same output (the zebra) using the same concepts, as proposed by Kim et al. (2018). We extract the concepts using images from the Broden dataset (short for broad and densely labeled), which was created by combining several semantic segmentation and classification datasets in order to dissect deep representations. The Broden dataset contains sufficient examples for each concept to be able to effectively distinguish the internal activations. We extract four different concept descriptions using images from the Broden dataset (Bau et al., 2017) to dissect the *zebra* classification into the concepts of *stripes*, *dots*, *zigzags* and an abstract representation of the horse-family concept *equidae* (horses,


$\mathcal{G}_\theta(\mathcal{K})$
$\mathcal{G}_\theta(\boldsymbol{equid}(x))$
$\mathcal{G}_\theta(\boldsymbol{stripes}(x))$
$\mathcal{G}_\theta(\boldsymbol{zebra}(x))$
Background Knowledge
Input Data
$\mathcal{K}$
$f_l : \mathbb{R}^n \to \mathbb{R}^m$
m
Pth class
trained CNN
Model Satisfiability: STOP if $\mathcal{G}_\theta(\mathcal{K} \cup \mathcal{K}_{new}) \approx 1$
Model Explanation to reason about what has been learned
If desired: add knowledge as new constraints $(\mathcal{K}_{new})$ for learning
Global
Local
Quagga

Query	Output	Formula
$\mathcal{G}(\phi_1)$	$sat_{\phi_1} = 0.96$	$\phi_1 = \forall x : zebra(x) \to stripe(x)$
$\mathcal{G}(\phi_2)$	$sat_{\phi_2} = 0.29$	$\phi_2 = \forall x : zebra(x) \to zigzag(x)$
$\mathcal{G}(\phi_3)$	$sat_{\phi_3} = 0.03$	$\phi_3 = \forall x : zebra(x) \to dots(x)$
$\mathcal{G}(\phi_4)$	$sat_{\phi_4} = 0.98$	$\phi_4 = \forall x : equid(x) \wedge stripe(x) \to zebra(x)$
		...

$\mathcal{G}(zebra(x_{qua})) = 0.99$
$\mathcal{G}(equid(x_{qua}) \wedge stripe(x_{qua})) = 0.97$

Figure 7.7: We produce local explanations (for individual inputs/images) and global explanations (universally-quantified formulas) for the deep learning model by querying specific neurons. We then reason about the generality of the explanations given the *truth queries* extracted from the trained network. The figure shows some of these queries associated with groundings in the neural network and their satisfiability (sat) levels. Using linear probes to ground the activation patterns of intermediate representations into the language of LTN, we are able to utilise abstract concepts as explanation symbols in the logic. Following querying, the neural model can be constrained based on a user selection of logical formulas $\mathcal{K}_{new}$ for further learning. This iterative process seeks to align the model with user values in the context of background knowledge $\mathcal{K}$. In the figure, the Quagga is classified as a zebra. A user's desire to change such classification should trigger the addition of knowledge into $\mathcal{K}_{new}$ informed by the queries to be satisfied by the final trained model. Notice that training from data may begin without any background knowledge which can be revised by querying user-defined concepts and constraints deemed as necessary for the network to learn.

donkeys, zebras and others). With the LTN integrated framework, the querying itself can be carried out independently of the data used during training. An entire dataset or, as in this case, an external dataset with previously unseen images can be fed as input to evaluate or retrain the model.

As outlined in the previous section, we learn a linear probe to ground the activation

patterns into the concepts specified to explain the prediction. Figure 7.4 illustrates how we extract the concept groundings from the feature mapping that gets fed into the GoogLeNet's fully connected classifier. When using mutually exclusive concepts to explain a model, it is possible to differentiate between linear probes by using a *softmax* activation. The linear probe will be able to draw counter-examples in the form of examples of alternative concepts or random images. In some instances, this may not be desirable, as concepts may be present simultaneously. In this scenario, we follow Kim et al. (2018) in selecting a negative set of images from the imagenet dataset, which are neither part of the class to be explained nor the concepts to be explained. We learn the groundings of the activation patterns for the specified concepts from 150 images of each concept and an equal number of negative examples for each class. Subsequently, the truth-value of each query is calculated through fuzzy logic inference using LTN. These queries can be specific to an image (local) or aggregated across the entire set of examples (global).
The quantifier $\forall$ is used to aggregate across a set of data points by replacing x with every image available from the dataset thus evaluating the model's behaviour across all available data. The following implication: $\forall x : zebra(x) \rightarrow stripe(x)$, with the symbol $stripe(x)$ being replaced by the corresponding concept grounding in the network, provides an insight onto how important the concept $stripe(x)$ is for the CNN's classification output $zebra(x)$ given the set of images x.

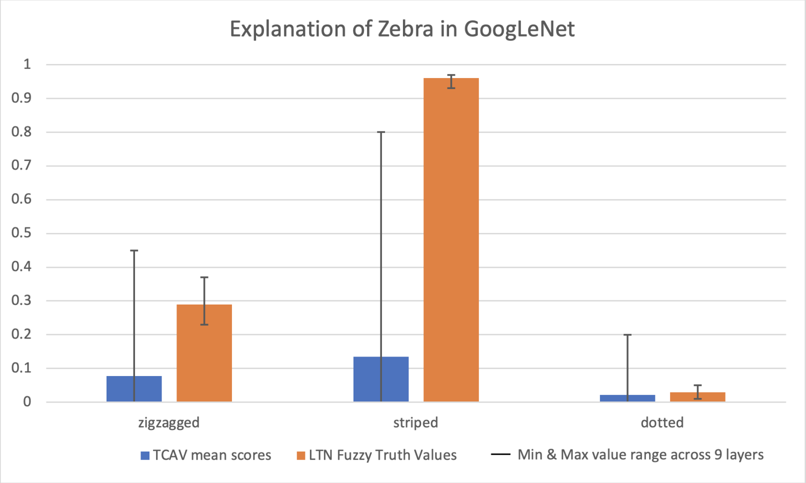


Figure 7.8: TCAV scores and fuzzy truth values for concept explanation of Zebra in GoogLeNet trained on imagenet for comparison. The coloured bar shows the mean value with the black line showing the range of the values.

The graph in Figure 7.8 shows the mean TCAV scores and mean fuzzy truth values for the three concepts across 9 different inception layers (3a, 3b, 4a, 4b, 4c, 4d, 4e, 5a, 5b). We calculate the concepts based on 30 test images of zebras. TCAV scores are directly computed and aggregated based on TCAV-specific definitions, whereas we utilise the predefined $\forall$ operator of fuzzy logic. The added benefit of grounding in fuzzy truth values is the relatively accessible interpretation such values offer compared to the TCAV scores, which are unique to the method. While the TCAV scores differ significantly between layers for the same concept, the fuzzy truth values are stable across layers.

Another important advantage of the fuzzy logic approach is the fact that we can combine several concepts using the logic: $\forall x : equid(x) \wedge stripe(x) \rightarrow zebra(x)$ returns a truth value of 0.98 across a set of 3000 examples from ImageNet, indicating that the CNN assigns any horse-like object with stripes to the class of zebras. When applying a universal quantifier, the user is able to evaluate the decision making process of the model in general, by examining the concepts on all available data, even if it has not been used for training, thereby producing a global explanation.

One example image previously unknown to the model is the extinct *quagga*, an animal characterised by a brown striped coat instead of the black and white pattern of zebras which has been selected to illustrate the potential of local explanations. The model identifies this animal correctly as a member of the *Equidae* family, recognises the stripes on the animal and consequently classifies the image as that of a zebra, as shown in Figure 7.7. By utilising the trained linear probes of the activation vectors to ground individual images, we generate local explanations that provide insight into why a particular image might be classified in a certain way according to the model.

Upon identifying potential undesired behaviour, for example by querying known exceptions, a user can add new rules into the knowledge-base (by adding logical formulas into $\mathcal{K}_{new}$) for further training of the network. In case the specification of quagga and zebra is to be changed, an alternative inference process can be imposed on the CNN model. Recall that quagga are currently considered by the CNN to be a subspecies of zebra. Assuming that the user decides to change this, as an example, let us consider introducing concept probes $bw(x)$ for *black and white* objects and $col(x)$ for *colourful objects*, and let us assume that these concepts are to be regarded as mutually exclusive. Adding the following rule to $\mathcal{K}_{new}$ as a new constraint to be satisfied by learning should force the neural model to only classify black and white objects as zebras: $\phi_5 = \forall x : equid(x) \wedge stripe(x) \wedge \neg bw(x) \rightarrow \neg zebra(x)$.[1]

Before further training, ϕ_5 exhibits a low *sat*-level of $sat_{\phi_5} = 0.09$, as the model classifies all objects associated with the $equid(x)$ and the $stripe(x)$ concepts to the *zebra* class regardless of their color. By retraining for only five iterations, the *sat*-level increases to $sat_{\phi_5} = 0.94$, which indicates that only black and white objects (in conjunction with the *stripe* and *equidae* concepts) are now considered to be zebras. Therefore, the example image of the quagga is no longer inferred to be in the zebra class with $\mathcal{G}(zebra(x_{qua})) = 0.08$, where x_{qua} denotes the image of a quagga. The neural model nevertheless identifies correctly the equidae and stripe concepts in the quagga, with $\mathcal{G}(equid(x_{qua}) \wedge stripe(x_{qua})) = 0.97$.

It should be noted that the explanation itself does not affect the performance of the model. Thus, prior to revising $\mathcal{K}$, the behaviour of the model remains unchanged due to the use of linear probes that solely interpret activation patterns. Kim et al. (2018) demonstrate that connecting output labels to model-inherent abstract rep-

[1] Notice that the satisfiability of this rule should be the same as that of $\forall x : equid(x) \wedge stripe(x) \wedge col(x) \rightarrow \neg zebra(x)$, as we apply a $softmax$ function to mutually exclusive concept probes; in this case $\forall x : col(x) \leftrightarrow \neg bw(x)$.

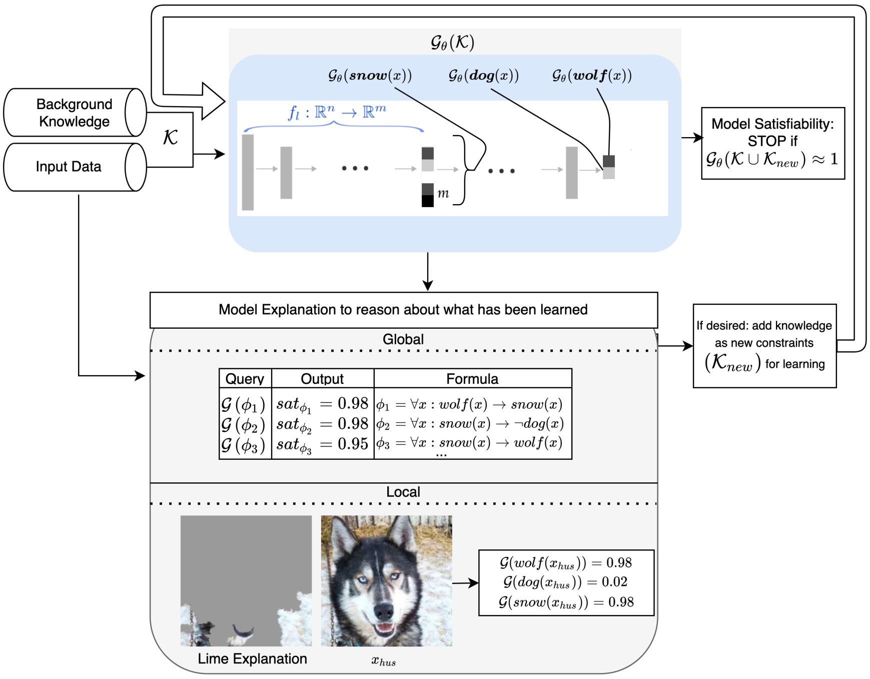
resentation is generalisable across a variety of computer vision tasks. This section includes a direct comparison that illustrates that such extraction using our method is significantly more stable than scores obtained by TCAV. A further advantage is the illustrated combination of abstract concepts to derive model output explanations and the ability to influence the behaviour of the model directly. Our approach and the TCAV method have the disadvantage that exemplary data must be collected by hand for each concept required for extraction, as well as the fact that the model has learned representations of concepts.

In the next scenario, we will recreate one of the most prominently presented examples in which a model induces undesirable correlations during training. Frequently cited (Ribeiro et al., 2016) and often referred to in introductory session on XAI, it illustrates how undesirable correlations can be picked up by the classifier because of flawed data collection. The identification of these types of anomalies can be a challenge from merely examining the training data and observed predictions for the classes, but XAI techniques can provide insight into potential undesired heuristics of network architectures.

We reproduce the experiment by Ribeiro et al. (2016) that proposed a classifier for distinguishing between images of *wolves* and *dogs* (huskies). We imitate the training process of Ribeiro et al. (2016) by training a logistic regression classifier on the features mappings of Google's pre-trained Inception CNN. The training set of 80 training images for wolves and huskies has been hand-selected. For pictures of wolves, we ensure that there is snow in the background, whereas for images of huskies, there is not. Through the use of this biased dataset, we make sure that the classifier consistently predicts *wolf* when there is snow or white surrounding the object, and *dog* otherwise, regardless of any complex features such as animal colour, pose, or fur. The classifier is intentionally trained to learn this simple predictive shortcut, as Ribeiro et al. (2016) shows that users stop trusting the classifier when they are provided with an explanation of the flawed internal model reasoning. In Ribeiro et al. (2016), explanations are more complex, since participants need to connect the highlighted pixel values in the image that identify significant parts of the image with the detection of snow as a potential feature (see Figure 7.9 bottom). We provide logical explanations in the form of FOL queries that can even be expressed in natural language.

To learn the linear probe, we utilise 25 images depicting nature with and without snow in order to ground the images into the predicates *snow*, *dog*, and *wolf.* We subsequently query the model on a validation set with 1000 imagenet examples that contain huskies as well as dogs. Through the use of quantification, we derive the global explanation that provides us with insights into the general model behaviour. The following queries illustrate how the *sat*-values can confirm that the model uses simple heuristics to arrive at its decision.

sat_{ϕ_1}, where $\phi_1 = \forall x : wolf(x) \rightarrow snow(x)$ confirms that across all images, the model recognises snow in images that are classified as wolf. Additionally, the queries indicate that images with snow are not considered dogs. The query ϕ_3 concludes that the recognition of snow coincides with the model recognising a wolf as expressed by its high satisfiability.


$\mathcal{G}_\theta(\mathcal{K})$
$\mathcal{G}_\theta(\boldsymbol{snow}(x))$
$\mathcal{G}_\theta(\boldsymbol{dog}(x))$
$\mathcal{G}_\theta(\boldsymbol{wolf}(x))$
Background Knowledge
Input Data
$\mathcal{K}$
$f_l : \mathbb{R}^n \rightarrow \mathbb{R}^m$
m
Model Satisfiability: STOP if $\mathcal{G}_\theta(\mathcal{K} \cup \mathcal{K}_{new}) \approx 1$
Model Explanation to reason about what has been learned
If desired: add knowledge as new constraints $(\mathcal{K}_{new})$ for learning
Global
Local
Lime Explanation
x_{hus}

Query	Output	Formula
$\mathcal{G}(\phi_1)$	$sat_{\phi_1} = 0.98$	$\phi_1 = \forall x : wolf(x) \rightarrow snow(x)$
$\mathcal{G}(\phi_2)$	$sat_{\phi_2} = 0.98$	$\phi_2 = \forall x : snow(x) \rightarrow \neg dog(x)$
$\mathcal{G}(\phi_3)$	$sat_{\phi_3} = 0.95$	$\phi_3 = \forall x : snow(x) \rightarrow wolf(x)$ ...

$\mathcal{G}(wolf(x_{hus})) = 0.98$
$\mathcal{G}(dog(x_{hus})) = 0.02$
$\mathcal{G}(snow(x_{hus})) = 0.98$

Figure 7.9: Fuzzy truth values for concept explanation of a husky in comparison with Lime (Ribeiro et al., 2016). In this example, the framework proves valuable as a means of confirming general patterns of decision-making where local explanations allow the discovery of specific actions.

Additionally, we depict the local explanation in order to provide a comparison to the LIME explanation (Figure 7.9 bottom). When we ground a particular image of a *husky* (x_{hus}), we can determine that the model recognises *snow* and classifies the animal as a *wolf*. Unlike the LIME explanation, which relies on the interpretation of the highlighted pixels by the user, our explanation grounds the image in human readable concepts. Thus, our method can be directly understood without the need for external interpretation.

Through the presented process, we wish to illustrate how alternative XAI approaches may offer valuable insights into a variety of concepts. In this regard, local explanations are particularly useful since they allow for more precise descriptions of individual images. This could be useful for raising awareness of conceptual and relational queries that may be of interest to users. LIME is limited to highlighting relevant regions in a particular image, however, we evaluate behaviour across the entire dataset. Nevertheless, the disadvantage is that the behaviour that requires evaluation must be specified. In such situations, the framework may prove valuable as a means of confirming general patterns of decision-making where local explanations allow the discovery of specific model behaviour.

The first experiment demonstrated how a robust model can be integrated into the framework and queried to enhance trust. Using inherent concept representations, we further illustrated how logical constraints could be used to influence this model's decision-making process. Additionally, we demonstrated how this approach could be employed to highlight flaws in model reasoning using a widely used XAI example. We outlined the use of a convolutional neural network in the first set of examples, and in the following section, we demonstrate the integration of the transformer architecture including image encoders.

7.4.2 Transformer & Encoder

We illustrate in the second set of examples that our method meets the requirements set by Doran et al. (2018) for *truly explainable AI.* The authors outline that existing approaches "are lacking in their ability to formulate, for the user, a line of reasoning that explains the decision making process of a model using human-understandable features of the input data." These human-understandable features in this context refer to abstract representations of the low-level data as outlined.

They argue that existing methods leave the explanation generation to humans depending on their background and claim that this exercise is dangerous as varying knowledge about the data and its domain will lead to the deduction of different explanations about why a model makes a decision. Furthermore, the authors propose that "reasoning is a critical step in formulating an explanation about why or how some event has occurred".

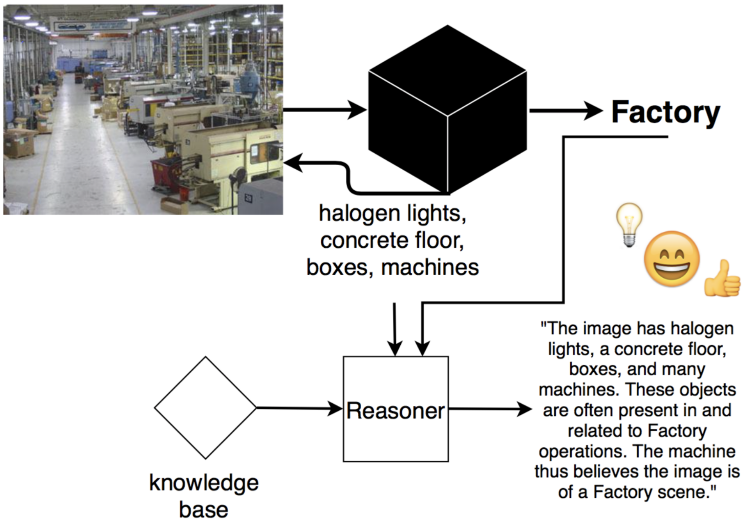

Factory
halogen lights,
concrete floor,
boxes, machines
Reasoner
knowledge
base
"The image has halogen lights, a concrete floor, boxes, and many machines. These objects are often present in and related to Factory operations. The machine thus believes the image is of a Factory scene."

Figure 7.10: Illustration of a "truly explainable AI" system according to Doran et al. (2018). Here the combination of symbols emitted in combination with background knowledge yields a logical deduction about their relationship in the system's decision making process.

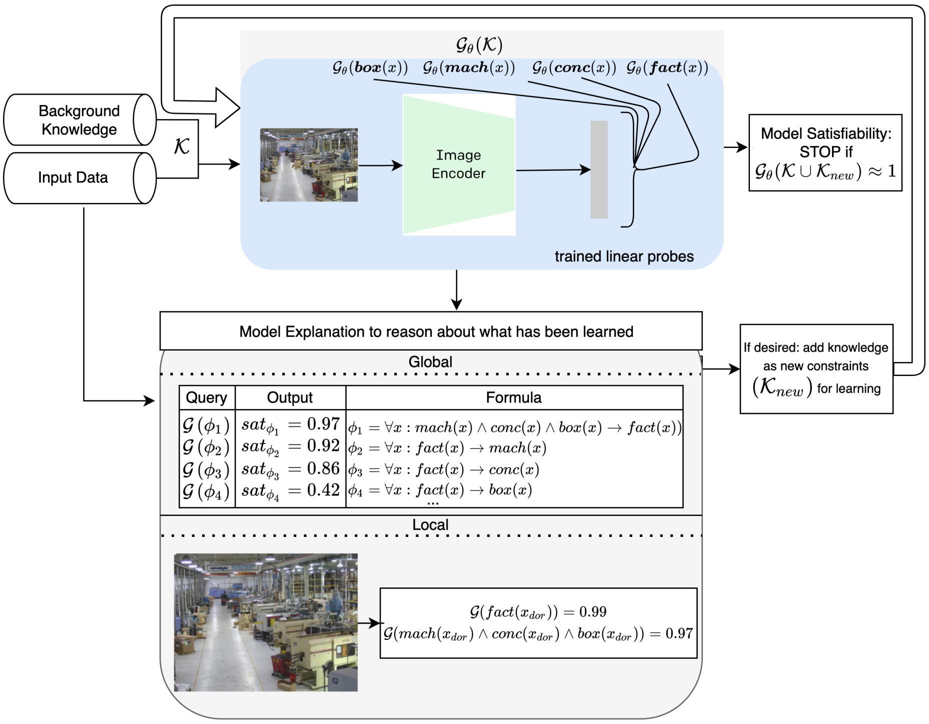
We demonstrate that we can reproduce the example provided byy Doran et al. (2018) and illustrated in Figure 7.10 with our system to show the adaptability of the approach. We use the pipeline depicted in Figure 7.5 that incorporates the CLIP image encoder that was trained on 400 million pairs of image and text Radford et al. (2021). These text pairs primarily contain descriptions of images in natural language which are therefore highly suitable for grounding the model's representation in concepts that can be understood by humans.

We collect images of *boxes*, *concrete*, *machines*, and *factories* using the imagenet database with 100 example images each and equally sized negative sets using random images as proposed by (Kim et al., 2018). Subsequently, we train linear probes utilising the extracted image embeddings vector representation by CLIP to ground the images according to their respective labels.

This allows one to generate the explanation specified by Doran et al. (2018) translated into first-order logic clauses. Through this method, we can generate local explanations that are based on explicit grounding of images, and global explanations that are based upon querying quantifiable variables across entire datasets.

In the explicit example image presented in Figure 7.10, the classifier grounding for the factory predicate returns a high truth value. Furthermore, the system recognises a machine, concrete, and lights in the image as intended (visualised in Figure 7.11)

To make more general statements about global model behaviour, we can use the



$\mathcal{G}_\theta(\mathcal{K})$
$\mathcal{G}_\theta(\boldsymbol{box}(x))$
$\mathcal{G}_\theta(\boldsymbol{mach}(x))$
$\mathcal{G}_\theta(\boldsymbol{conc}(x))$
$\mathcal{G}_\theta(\boldsymbol{fact}(x))$
Background Knowledge
Input Data
$\mathcal{K}$
Image Encoder
trained linear probes
Model Satisfiability: STOP if $\mathcal{G}_\theta(\mathcal{K} \cup \mathcal{K}_{new}) \approx 1$
Model Explanation to reason about what has been learned
If desired: add knowledge as new constraints $(\mathcal{K}_{new})$ for learning
Global
Local
$\mathcal{G}(fact(x_{dor})) = 0.99$
$\mathcal{G}(mach(x_{dor}) \wedge conc(x_{dor}) \wedge box(x_{dor})) = 0.97$

Query	Output	Formula
$\mathcal{G}(\phi_1)$	$sat_{\phi_1} = 0.97$	$\phi_1 = \forall x : mach(x) \wedge conc(x) \wedge box(x) \rightarrow fact(x))$
$\mathcal{G}(\phi_2)$	$sat_{\phi_2} = 0.92$	$\phi_2 = \forall x : fact(x) \rightarrow mach(x)$
$\mathcal{G}(\phi_3)$	$sat_{\phi_3} = 0.86$	$\phi_3 = \forall x : fact(x) \rightarrow conc(x)$
$\mathcal{G}(\phi_4)$	$sat_{\phi_4} = 0.42$	$\phi_4 = \forall x : fact(x) \rightarrow box(x)$ ...

Figure 7.11: We illustrate that we can produce the desirable explanation proposed by Doran et al. (2018). We incorporate the CLIP image encoder by Radford et al. (2021) and collect images of boxes, concrete, machines, and factories using the imagenet database with 100 example images each. Subsequently, we train linear probes utilising the extracted image embedding vector representations by CLIP to ground the images according to their respective labels. Subsequently we use fuzzy logic to evaluate the reasoning process by incorpoating the extracted concepts.

quantifiable formulas $\phi_1 = \forall x : mach(x) \wedge conc(x) \wedge box(x) \rightarrow fact(x)$ evaluated on 2000 images extracted from imagenet to interpret the grounding $\mathcal{G}_{\phi_1} = 0.97$. The model classifies a factory whenever it observes machines, boxes, and concrete in the sampled set. It is also possible to switch the antecedent and consequent in order to reason about a possible bi-implication as learned behaviour. The grounding of $\mathcal{G}(\phi_2) = \forall x : fact(x) \rightarrow mach(x) = 0.92$ indicates that for the factories that are recognised, the model does not implicitly identify machines, and therefore does not take advantage of them as shortcuts for the classification of *factories*. Furthermore, the model also classifies factories that do not have concrete floors, manifested in the following query: $\mathcal{G}(\phi_3) = \forall x : fact(x) \rightarrow conc(x) = 0.86$. Boxes seem to be the least important as shown by the small *sat*-level for: $\mathcal{G}(\phi_4) = \forall x : fact(x) \rightarrow box(x) = 0.46$. There is no restriction on the number of concepts, combinations, and logical formulae. We can use first-order logic to derive explanations and reason about what has been learned as the user desires.

Looking at the image of the Doran et al. (2018) example explicitly, we can conclude that the model indeed does recognise boxes, concrete, and machines. As such, it recognises the image as factory, as it has done for every image within the dataset that contains all of the concept attributes in conjunction.

In the next example, we will demonstrate that by utilising the full CLIP transformer including text encoder, we are able to learn all desired groundings using a zero-shot methodology. The desired concepts that need grounding are presented as a natural language input, allowing for a determination of groundings without the need for exemplary data, but only a description of the concept in natural language. In this approach, groundings will be derived based on the transformer embeddings by calculating the cosine similarity of the concept prompts and encoded image and feeding this value into a *softmax* function. We subsequently integrate these groundings into the LTN framework for First-Order-Logic querying, as illustrated in Figure 7.12.

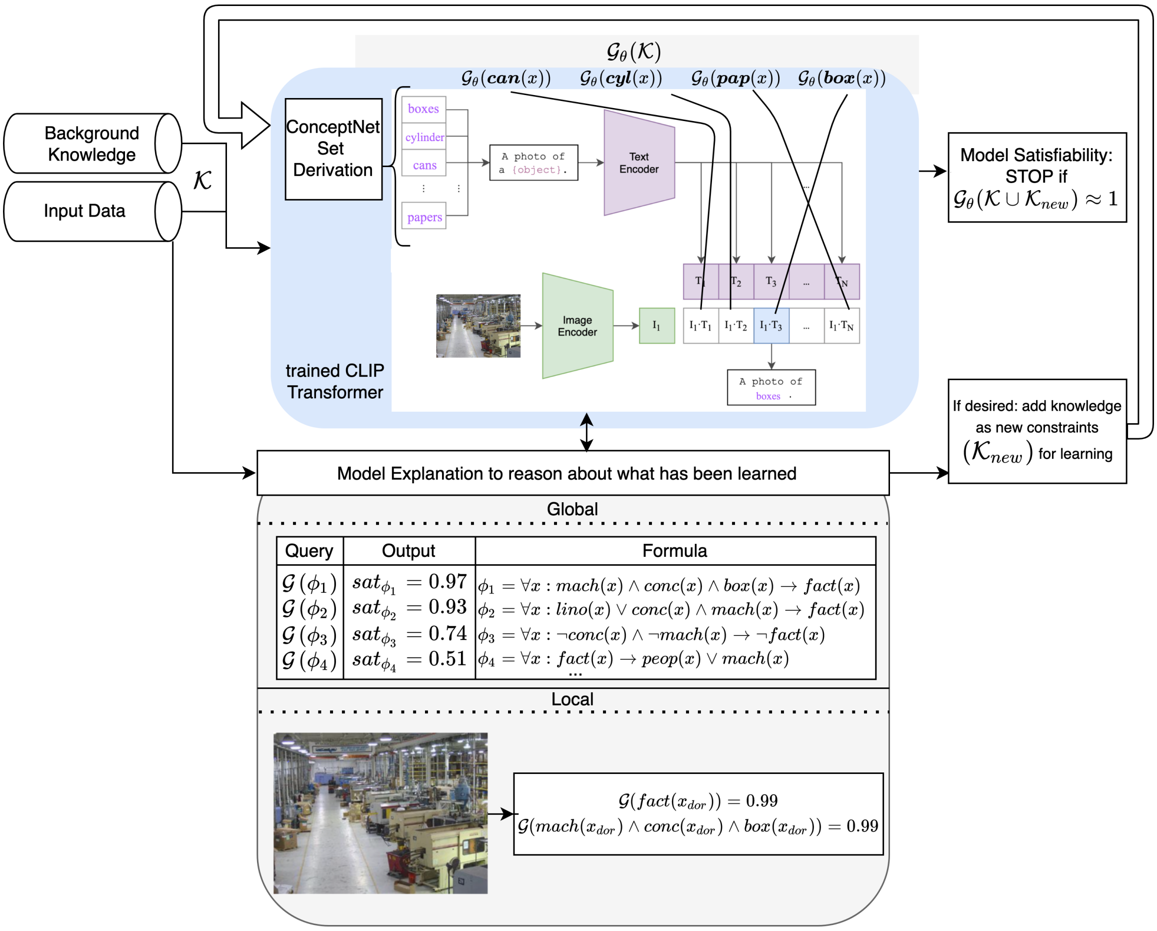
The groundings based on CLIP embeddings can only be derived on a zero-shot basis, however, only if the prompts are mutually exclusive. Rather than being generative, this model is discriminative, which means that it can only derive the most likely description given a given set of prompts.

By utilising the external knowledge graph ConceptNet as part of the pipeline, a user is only required to provide a concept of interest in order to generate a set with mutually exclusive concepts automatically. ConceptNet is a multilingual knowledge graph aimed at mapping the meaning of words that people use. The semantic network is utilised here to derive counter-sets to each prompt using the ConceptNet attributes *antonym* and *distinct terms*.

In our factory example, the user may specify the prompt "A photo of a concrete floor" as the descriptive abstraction of interest. ConceptNet finds counter-examples by replacing the concept "concrete" with "wooden", "tiled", and "linoleum floor" in the pipeline. Similarly, negative set derivations are done for the prompt "A photo of boxes" being replaced by "cylinders" "bins", and "papers". "A photo of a factory" will have the negative set of "artist", "farm", and "company". For the concept of "machine", ConceptNet derives "human", "people", and "organic things".

In the event that ConceptNet fails to find terms that appear reasonable, the user may manually refine the derived prompts. Moreover, the hyper-parameter responsible for generating four alternative prompts for each of the concepts may also be changed.

The global explanation is in line with the previous explanation as the query: $\phi_1 =$


Background Knowledge
Input Data
$\mathcal{K}$
ConceptNet Set Derivation
boxes
cylinder
cans
papers
A photo of a {object}.
Text Encoder
$\mathcal{G}_\theta(\mathcal{K})$
$\mathcal{G}_\theta(\boldsymbol{can}(x))$
$\mathcal{G}_\theta(\boldsymbol{cyl}(x))$
$\mathcal{G}_\theta(\boldsymbol{pap}(x))$
$\mathcal{G}_\theta(\boldsymbol{box}(x))$
T_1 T_2 T_3 ... T_N
Image Encoder
I_1
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
A photo of boxes .
trained CLIP Transformer
Model Satisfiability: STOP if $\mathcal{G}_\theta(\mathcal{K} \cup \mathcal{K}_{new}) \approx 1$
If desired: add knowledge as new constraints $(\mathcal{K}_{new})$ for learning
Model Explanation to reason about what has been learned
Global
Local
$\mathcal{G}(fact(x_{dor})) = 0.99$
$\mathcal{G}(mach(x_{dor}) \wedge conc(x_{dor}) \wedge box(x_{dor})) = 0.99$

Query	Output	Formula
$\mathcal{G}(\phi_1)$	$sat_{\phi_1} = 0.97$	$\phi_1 = \forall x : mach(x) \wedge conc(x) \wedge box(x) \rightarrow fact(x)$
$\mathcal{G}(\phi_2)$	$sat_{\phi_2} = 0.93$	$\phi_2 = \forall x : lino(x) \vee conc(x) \wedge mach(x) \rightarrow fact(x)$
$\mathcal{G}(\phi_3)$	$sat_{\phi_3} = 0.74$	$\phi_3 = \forall x : \neg conc(x) \wedge \neg mach(x) \rightarrow \neg fact(x)$
$\mathcal{G}(\phi_4)$	$sat_{\phi_4} = 0.51$	$\phi_4 = \forall x : fact(x) \rightarrow peop(x) \vee mach(x)$ ...

Figure 7.12: We use the explanation benchmark by Doran et al. (2018) to derive zero-shot explanations. Here the user only has to give the natural language description of the concepts of interest as we use ConceptNet to derive mutually exclusive concepts. These descriptions are encoded along the image of interest to derive explanations without the necessity to collect example images for the concepts of interest. The cosine similarities are then used to ground the concepts into the logical clauses to derive model explanations.

$\forall x : mach(x) \wedge conc(x) \wedge box(x) \rightarrow fact(x)$ also returns a truth value of 0.97. This means on a test set of 2000 images, whenever the model recognises machines, concrete, and boxes it infers that the image shows a factory.

Moreover, we can derive interesting observations from the generated negative sets as a result from parsing ConceptNet. Here, the generated negative set included the concept of linoleum flooring which is a reasonable alternative for factory floors.

By using the CLIP model's local explanation for the explicit example image provided by Doran et al. (2018), we can discover that the model is indecisive regarding the floor texture ($\mathcal{G}(lino(x_{dor})) = 0.68$ and $\mathcal{G}(conc(x_{dor})) = 0.32$). However, the model identifies boxes and machines, and consequently infers a factory with high confidence.

By evaluation across the test set, we can query: $\phi_2 = \forall x : lino(x) \vee conc(x) \wedge$

$mach(x) \rightarrow fact(x)$ which indicates that when the model detects a machine on a concrete or linoleum floor, it also classifies that image as a factory. In addition, we can employ negation and inquire whether the absence of concrete or a machine necessarily implies the model does not recognise a factory ($\mathcal{G}(\phi_3) = \mathcal{G}(\forall x : \neg conc(x) \wedge \neg mach(x) \rightarrow \neg fact(x)) = 0.74$). We may also investigate whether the model considers that the factory class implies either humans or machines being present in the image ($\phi_4 = \forall x : fact(x) \rightarrow peop(x) \vee mach(x)$).

However, CLIP still presents some limitations that directly affect our integration. Although the transfer learning capabilities offer an improvement over the standard CNN architecture, zero-shot performance on particular datasets remains below that of task-specific models. It is particularly true for more fine-grained classification, such as identifying species of flowers or variants of aircraft. Despite the large sample size used for pre-training generalizing well for some natural distributions, Radford et al. (2021) demonstrate that out-of-distribution generalization still remains poor. In addition, CLIP is limited to choosing only from a given set of concepts in a given zero-shot classifier, despite our attempts to expand the scope by integrating an external ontology. Compared to a more flexible approach such as image captioning which could produce output, this is a significant restriction.
It is also important to consider the nature of the training data. CLIP learns to ignore properties and objects that are ignored in captions. In essence, it allows the model to learn invariance rather than imposing it through its architecture or by constructing transformations for data augmentation. The data curation process defines this invariance, so the way we choose our noisy labels or captions is the key to determining how robust the model will be. The implications of this are significant in domains in which all objects within a frame are critical, such as autonomous vehicles. As a result, we propose that our tool may be increasingly useful in calling attention to specific model behaviors and revealing ignored concepts in labels or invariances in fine-grained classifications.

Using the experiments above, we wish to demonstrate the flexibility with which different model architectures can be integrated into the framework. This section emphasises that the approach is also capable of generalising to multi-modal data. Furthermore, the transformer integration enhances accessibility as the user is only required to present concepts that are of interest without the need for additional data collection as the zero-shot method is readily available. The study also showed how the generation of sets could be enhanced by external knowledge graphs so that the user does not have to provide alternative concepts manually. Additionally, the experiment illustrates the flexibility with which different types of concepts can be extracted and combined using fuzzy logic. Since we can ground single images as well as aggregate using quantifiable axioms, we are able to reason about the general decision-making process the model has learned as well as why a particular decision was made.

7.5 Summary of Our Approach to Interactive XAI

Our work outlines the adaptation of the LTN framework to facilitate intuitive model explanations based on abstract representation and interactive querying by using first-order logic queries to check for knowledge learned by the deep neural network. Our proposal includes three distinct contributions. We first present a definition for practical reasoning and evaluate the method's ability to reason about what has been learned. Second, we evaluate how the framework can be utilised to discover and refine models that yield unfair classifications based on biased datasets. Third, we demonstrate how concept extraction and logic integration can be fully integrated to assist with explanation and revision of models.

The methodology section of this approach is a critical component of establishing a strong foundation for understanding the reasoning capability of neural networks through the use of differentiable fuzzy logic. The ability to reason about what is learned relies on the limitations of the fuzzy logic systems underlying all explanations. This evaluation focuses on assessing the reasoning capabilities of the proposed framework on the basis of datasets and identifying what background knowledge contributes to better deduction. As we intend to introduce a continuous mechanism for bidirectional communication, we showed how effectively the network can learn such constraints.

Based on our thorough evaluation of the limitations of differentiable fuzzy logic systems for extracting and refining knowledge continuously, we tested their suitability in domains common to XAI applications. The presented work outlines a method of preventing the models from learning unwanted behaviours and biases by acting on information extracted from XAI approaches. As a result of continuous constrained-based learning, our results demonstrate that by incorporating knowledge extraction from deep networks into the LTN framework and utilising tailored fairness constraints, it is possible to impart fairness into deep networks in a general manner. One of the key differences between our method and other XAI-based approaches is evident. The fairness experiments demonstrate that simply identifying undesirable model behaviour is of limited value in certain situations. We show that our method can be utilised to identify and correct undesired model behaviour by leveraging another XAI method, in this case SHAP (Lundberg and Lee, 2017). On the basis of experiments conducted on three real-world data sets used to predict income, credit risk, and recidivism, we found that our approach can satisfy fairness metrics while maintaining state-of-the-art classification performance.

Along with incorporating bidirectional communication into the approach for achieving fairness, we demonstrated how Neural-Symbolic integration can be used to better understand the model's decision-making process. As the TCAV strategy is integrated with the LTN framework, complex conceptual explanations can be provided in an interactive learning environment. This framework is shown to be capable of decomposing models into explanations that employ complex concepts grounded in logic while facilitating user interaction. The method allows domain experts to study the data-inferred decision making process of complex deep neural networks by querying the model and refining the constraints in an iterative manner. We began by discussing

the possibilities of internal representations in the context of different model architectures that facilitate reusable concepts. For example, feature maps of convolutional networks and large transformer models are rich representations for model dissection, that can be used across a wide range of datasets.
In addition, we demonstrate how local XAI-methods may complement the framework by guiding particular queries. In some cases, the user may be unaware of a particular behaviour and may not consider querying it. It may be desirable to employ an XAI method in such scenarios as a guide by making use of lower-level information, which is normally restricted to local explanations. It is then possible for the user to inquire about a global behaviour by exploring how the model makes use of different concepts and representations to reach its conclusions.
Our research indicates that our method is capable of generating descriptions based on a semantic knowledge graph, such as ConceptNet, to provide a broad range of possible explanations for image classification. In this scenario, the user may specify a particular concept of relevance, and the pipeline will then generate descriptions based on the ConceptNet ontology. Our work has demonstrated that it is consistent with the standard established by Doran et al. (2018) for *truly explainable AI*. When using fuzzy logic to link symbols that represent abstract concepts, deep learning models can be understood in a manner similar to human explanations.

7.5.1 Interpretation of Fuzzy Logic

One observation is that the integration of fuzzy logic is more precise than other approaches to XAI. It is the outcome of fuzzy operators which satisfy certain mathematical properties. When compared to other XAI methods, such as TCAV, this represents a significant improvement because it establishes a standard. It is common for method-specific metrics to be introduced that may lead to confusion. As an example of this, SHAP values are frequently interpreted in an overly simplistic manner, ignoring the fact that these values are to be understood in relation to a specific baseline. In some instances, this premise may not be appropriate since the method can produce unrealistic baselines.

Moreover, the discretisation of symbol grounding exhibits some of the same characteristics. Language encourages us to delineate concepts and objects in a distinctive discrete manner for communication. At the same time, we recognise that reality is composed of concepts that overlap in nature. In addition, these fluid concepts are formed from low-level representations. Low-level distributed data is abundant, whereas discretely structured and clean data is rare. A neural network is more effective at learning when transforming low-level data into distributed representations, as opposed to localised representations.

Considering the form of representation, we are of the opinion that computation should remain sub-symbolic. In this thesis, we propose that Neural-Symbolic integration encompasses symbolic representation as an interpretation of underlying distributed representation. As opposed to the more traditional approach (Manhaeve et al., 2018), the neural and symbolic components do not constitute two separate entities but rather different abstractions of the same. Recent research on the computational origin of representation reinforces this assertion (Piantadosi, 2021). The author supports the notion that sub-symbolic dynamics may be used to generate higher-level representations of structures, systems of knowledge, and algorithmic processes. The author asserts that symbolic meanings and relations are the outcome of sub-symbolic dynamics that implement them. Furthermore, it is essential to recognise that a symbol's meaning is intrinsically linked to its use. Having followed this line of thought, it seems evident that the construction of new symbols can only be derived from the construction of a semantically meaningful space. It is our intention to depict this in a vector space where concepts are interrelated and may overlap.

The examples should have made it clear that even if tools are available for aligning representations and operations, bridging the gap between low-level information and abstract cognition will not be straightforward. The simplification of complex processes and distributed representations in order to make them human comprehensible will always be a trade-off that needs to be considered and balanced. The Neural-Symbolic integration, however, allows us to be considerate of more underlying formalisations of cognitive processes in humans and how they may be best represented. Consequently, in the context of machine learning we may ask whether we want the model to portray the world as it is or how humans perceive it. Is the long-term objective of artificial intelligence to produce intelligence on a level that is comparable to human intelligence, or is it to surpass human intelligence?

Chapter 8

Conclusion

In this thesis, we explored the Neural-Symbolic approach to explainable Artificial Intelligence. Traditionally, explainability has been considered merely a consequence of integrating neural and symbolic systems. Further, many XAI methods have been proposed without taking into account the extensive research conducted on Neural-Symbolic AI. With this work, we propose to bridge the gap between neural and symbolic paradigms to enhance understanding of model behaviour, but also between explainable AI and Neural-Symbolic AI, to influence the observed characteristics of the system interactively.

Initially, this thesis explored various explainability methods to determine whether any promising approaches could be deemed Neural-Symbolic in their ability to allow one to reason about what the model has learned. Through this work, we have compiled a novel taxonomy for explainable artificial intelligence that facilitates an improved practical overview by categorising a wide variety of approaches in an organised way, and that promotes a link with the structure of earlier work on rule-extraction methods by Andrews et al. (1995). The taxonomy also lends itself to the extension of further XAI and Neural-Symbolic methods in the future. The identification of gaps and diverging trends between Neural-Symbolic approaches and contemporary XAI approaches led us to investigate two promising approaches in greater depth. Specifically, these methods are designed to follow the inductive biases of neural networks more closely by utilising the fundamental components of internal representation and operation.

In our initial experiments, we focused on the TCAV method because, as a form symbol grounding, it enables the extraction of concepts as fundamental building blocks for model explanations. A key advantage of this approach is that it presents the model explanation through comprehensible symbols rather than emphasising the relevance of specific features, which could result in the identification of specific pixel values or word vector measurements. We were able to successfully associate the predictions of the model with abstract concepts in this manner. In the course of connecting the importance of different concepts to various output categories, the end user may be able to assess the appropriateness of the model.

We developed the proposed method further by applying the idea of Kim et al. (2018) to the natural language domain. By utilising ontology queries, the approach is improved subsequently in order to arrive at more comprehensive and semantically rich explanations.

Despite this, we encountered some significant limitations with these approaches. A potential challenge might be the interpretation of the TCAV scores, which may not be as straightforward as originally intended by the authors Kim et al. (2017). Despite the fact that the normalised scores give insight into the importance of concepts, they are not intended to adhere strictly to any mathematical properties or logic. As a result, the interpretation is based upon relating the TCAV scores to other concepts or output classes.

Most notably, the importance of concepts can only be assessed individually, and the user is not provided an opportunity to connect concepts in order to understand the model inherent relations. As an example, sarcastic undertones may be used to discuss a particular concept when a person struggles with mental health. A classifier that would learn this combination of multi-modal concepts would require the XAI method to account for the combination of concepts to be explained appropriately. As a result, isolating the significance of particular abstractions, while ignoring the internal operations of the model tying them together, leaves out an essential component that should be addressed in model explanations.

We then examined whether it was feasible to distill complex models into a simpler form in order to be able to overcome the black-box limitations and learn about the model-inherent operations. By translating the model into a decision tree we mimic the hierarchical and sequential flow of information in neural networks. Through the construction of a decision tree, we are capable of understanding the operations within a model and provide the user with the ability to trace the decision-making process.
In this thesis, we presented an adaptation of the distillation approach using Soft Decision Trees to increasingly complex visual tasks. As a result of constraining the model by forcing it to commit to the most probable path, we can trace the decision-making process throughout the tree, which facilitates better understanding of the model. The distillation process is capable of producing interpretations that can be considered to be understandable from an outside perspective, and, therefore, be compared to traditional Knowledge Extraction methods according to Brachman and Levesque (2004). However, the proposed approach of the model did not allow the integration of updated knowledge back into the network once we have gained insight into the model's behaviour. In accordance with Bader and Hitzler (2005), it should be desirable for knowledge to be extracted and revised, but there is no mechanism for translating revised knowledge back into the model. Nevertheless, this limitation also affects some traditional rule-extraction models such as TREPAN.
We also observed significant limitations as a result of the complexity of the task. Node masks are incapable of learning abstractions, and convolutional kernels cannot be utilised. This drastically decreases this method's effectiveness in domains with low-level data that require abstraction in order to be effective. Further, the expressiveness of the operations is limited as compared to logic-based methods. Despite the fact that a soft decision tree representation is fully differentiable, it lacks the power of logic in terms of precision. As a result, both the extracted representations as well as the relationships among these representations impair an attempt to approximate

a black-box model as closely as possible.

As a means of achieving intuitive and compelling model explanations, XAI methods may be required to allow for representations that capture powerful concept representations based on data integrated with comprehensive operations. The development of a powerful yet intuitive approach to XAI may be dependent both on the ability to extract the inner representations of concepts as well as the underlying operation in which the concepts are embedded.

A study of the TCAV approach indicated that it is effective at extracting representation and performing links between the output of the model and high-level concepts. Explanations at the concept level allow for intuitive explanations, but do not afford the opportunity to grasp the underlying relationships between these concept symbols in the context of model reasoning. The study of Soft Decision Trees indicates, however, that methods that allow full transparency into the model reasoning process may not be able to handle complex tasks. Presently, most XAI methods are highly specialized in that they focus on a particular aspect of the machine learning pipeline rather than viewing it as an effective communication channel more broadly.

Due to observing fundamental limitations in current XAI approaches, we believe that the most promising form of explanation in AI is the integration of symbolic representations and their inherent logic, in conjunction with the powerful learning capabilities of neural networks. In high-dimensional domains such as images and videos, logic explanations may be effective when applied to the appropriate level of abstraction, such as objects and their relationship, instead of pixel values.

The ultimate goal of a Neural-Symbolic approach to explainable AI system is the ability to enable bi-directional communication between the AI system and humans by acting as a lingua franca. This is an essential component for the achievement of trustworthy and reliable systems and requires the abstraction to human-like concepts (Gutzwiller and Reeder, 2021).

8.1 Neural-Symbolic Explainability & Future Work

The use of neural-symbolic methods is poised to be increasingly important in providing greater transparency to Artificial Intelligence. In this capacity, these methods are likely to serve as more than just another explainability approach.

As Machine Learning methods are becoming more widely used, it will be essential to provide explanations with varying degrees of abstractness and complexity, as outlined in Futia and Vetrò (2020). As part of our approach to XAI, we have attempted to present explanations that would enable laymen to to understand the underlying model representations.

The integration can be viewed as a paradigm shift allowing for active and interactive engagement between system and user by enabling communication. Therefore, the neural-symbolic integration serves as a common layer of communication in which different levels of abstraction can be utilised to exchange information from the model to its human counterpart and vice versa. The alignment of model and human reasoning with the objective of facilitating better understanding will enable us to increase the trust we have in models.

In our method, we attempt to address both the grounding and the relational components of symbolic systems. An integration of knowledge and data into a neural network, with the ability to extract post-hoc information and perform continual enhancement based on knowledge revision, is proposed as a bottom-up Neural-Symbolic approach to explainability.

As part of this approach, we provided three high-level contributions. Firstly, we presented a methodology and definition of practical reasoning to evaluate the method's capability to reason about what has been learned. Additionally, we analysed how the framework can be applied to the discovery and refinement of models that yield unfair classifications based on biased datasets. Thirdly, we demonstrated how concept extraction and logic integration can be fully integrated for model explanation and revision.

We assessed the reasoning capabilities of the proposed framework based on datasets and identify what information constitutes valuable background knowledge that contributes to better deduction. In particular, it was demonstrated that it is essential to consider whether a closed- or open-world assumption is appropriate for a given task. We demonstrated that a Neural-Symbolic system can be enhanced through the addition of redundant logic knowledge relevant to the influence of particular properties which are discovered through interactive queries.

Having thoroughly evaluated which limitations may be present in differentiable logic systems used to extract and refine knowledge continuously, we tested their practicality in common XAI application domains. One of the key differences between our approach and other XAI-based methods stands out from the experiment. We demonstrate through the fairness experiment that simply identifying undesirable model behaviour in certain situations is of limited value. Using the LTN framework, we discussed how to identify potential unequal treatment and illustrated how fairness may be achieved by using constraints in order to effectively minimise biases. We conducted experiments on three real-world data sets used to predict income, credit risk, and recidivism and found that our approach can satisfy fairness metrics while maintaining state-of-the-art classification performance. Comparing our method with other fairness-inducing ML techniques supports this conclusion.

The results are encouraging and indicate that fairness may be achievable in a flexible, model-agnostic manner. In order for fairness and equality to be a prominent consideration in Artificial Intelligence, we must provide practitioners with tools that make it easier to incorporate it into existing workflows.

As well as incorporating bidirectional communication into the approach to achieve fairness, we demonstrated how Neural-Symbolic integration can be used to better understand the model's decision-making process. Through the integration of the TCAV strategy with the LTN framework, we are able to provide complex conceptual explanations in an interactive learning environment. The framework is shown to be capable of decomposing model explanations into complex concepts and facilitate user interaction. Using it, domain experts can learn about the data-inferred decision making process of complex deep neural networks by querying the model and refining constraints for further learning, as part of an iterative process.

We presented a comparison between the fuzzy logic based extraction of truth values and TCAV methods in our analysis of neural-symbolic XAI and showed that the fuzzy logic properties of our systems present more consistent values with reduced variability across layers. Moreover, we demonstrated that this method can yield local as well as global explanations. XAI methods do not typically offer this level of flexibility, as shown in the overview table of chapter 3 Fuzzy logic ensures that both local and global explanations are always consistent with one another. This unification of explanations constitutes a promising direction for XAI. Based on the audience requirements, explanations can be tailored to meet the needs of different target groups. Level of abstraction and level of explanation of concepts may both be considered in the selection of the appropriate explanation. Nevertheless, it should be emphasised that meaningful explanations provided by this framework are dependent upon the representational and operational capability of the underlying model. Without any model-inherent form of conceptual representation, we are unable to combine any meaningful symbols to ground an explanation. Furthermore, we demonstrated how alternative XAI-methods can be incorporated into the interactive framework.

In this thesis, three distinct approaches were presented for the concept groundings, including a CNN and two transformer-based approaches. The CNN approach is designed to act as a blueprint for various different domain-specific adaptations. Here, the key proposition is that the method can be customised for different types of data and applications. As an example, the abstract concepts that are desired in the application of medicine may be subject-specific and domain dependent. On the other hand, transformers could be viewed as more general-purpose approaches to classification and caption that become intepretable using our method. Due to the fact that these models have been trained on an extensive variety of datasets, and in the case of CLIP on image captions, a large number of common concepts are likely to have already been captured by the models. This means that the representations are optimised to reveal various shapes, colours, and other common visual concepts that we may use for explaining the decision-making process of image classifiers.
By doing so, we were able to avoid the necessity of providing data samples to extract concepts. Due to the multi-modal nature of the model using natural language image descriptions, we may use the text encoder to provide a desireable description of the concept depicted in the image.
In terms of providing information-rich data representations, both the image encoder and the multi-modal transformer have proven to be highly effective and robust. Therefore, a user may modify the existing model to fit a particular use-case, while maintaining the ability to understand how the model makes decisions in the particular domain. Our research showed that the method is capable of generating alternative descriptions using a semantic knowledge graph, such as ConceptNet, in order to provide a broad range of possible captions for images. Users are offered an approach that provides a more accessible solution in which they are only required to provide a simple concept description appropriate to explain their application comprehensively. One of the advantages of using this method is that previously unknown facts are possibly revealed.

Our work has demonstrated that it satisfies the standard established by Doran et al. (2018) for *truly explainable AI*. By linking symbols that represent abstract concepts according to fuzzy logic, deep learning models can be understood in a way similar to explanations that would be provided by a human. Questions that may be posed in natural language from a model should simply be transformed into first-order logic before being assessed. Furthermore, it integrates reasoning into the explanation, which "is a critical step in formulating an explanation about why or how some event has occurred" Doran et al. (2018). We propose that our approach exceeds this benchmark by making the flow of information bidirectional and redefining the neural-symbolic interface as a common layer of communication.

Developing models that can yield generalisable concepts will be an ongoing effort. With increasing complexity and size of models, it appears that models are able to construct more complex transferable representations (see cross-dataset performance of CLIP in Figure 7.2). We consider the multi-modality of data to be an important step in the right direction as image captions might be able to convey a lot more nuance in describing the content of the depicted images. The use of natural language caption has led to a significant breakthrough in learning anthropocentric concepts for perception tasks, which needs to be further improved and integrated into a neural-symbolic framework for evaluation and explanation.

By providing a deeper and more comprehensive set of images or words relevant to the concept, it may improve concept grounding in the convolutional neural network and the transformer.

Furthermore, we believe that exploring the simultaneous learning of concepts and tasks may be of interest. In our research, we focused mostly on post-hoc extraction of information to reason about what has been learned and what needs to be addressed. Nevertheless, it may be worthwhile to enforce discrete representations into the model that will also be used to facilitate the execution of the intended task at the outset. As the loss is modular, we would be able to train the inner representation of the model by incorporating both the desired concepts as well as task-specific classification loss, for instance. Although this may result in a more transparent model architecture, it may limit the effectiveness of the model with respect to performance. However, some domains may benefit from these types of representations. As a result, direct rule extraction from localised neurons may become feasible (Odense and d'Avila Garcez, 2020).

As a result of the work by Kim et al. (2018) and Odense and d'Avila Garcez (2020), we decided to extract concepts at particular layers. In light of the fact that we are probing the network for the grounding of concepts, this may be extended to several layers across the network. The probe could theoretically monitor all layers or multiple layers within the network, whereas using the TCAV method is always limited to a single layer due to the underlying nature of the Concept Activation Vector.

Appendix A

Implementation of the LTN framework

We have developed a Python implementation of the interactive framework, which may be accessed at the following URL: https://github.com/benediktwagner/nesyxai. In this implementation, Real Logic formulas are converted to PyTorch computational graphs. Formulae of this type are capable of expressing complex statements about the data, prior knowledge that must be satisfied during learning, and statements which need to be proved. Additionally, this repository contains tutorials that effectively represent and handle the most important tasks associated with deep learning. Typical tasks of this type include classification, regression, clustering, and link prediction. In addition, the repository explains how to incorporate existing models in the framework. Similarly, we introduce the fairness example described in chapter 6 to demonstrate how fair classification can be achieved by using the framework and FOL constraints. The concept grounding work is also demonstrated using an accompanying notebook. We have further added the ability to utilise a GPU which greatly improves the efficiency of the method on large datasets.

Our continuous approach has also been incorporated into the main LTN library for Tensorflow, enabling a broader set of users. The repository can be found using the following URL: https://github.com/logictensornetworks/logictensornetworks). While the earlier version was limited to static computation, the current version is capable of interactive sessions (e.g. jupyter notebooks) which cache the parameters in order to incrementally add logic formulae to the knowledgebase $\mathcal{K}$.

Bibliography

A. Adadi and M. Berrada. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018. ISSN 21693536. doi: 10.1109/ACCESS.2018.2870052.

J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, 2018.

A. Agarwal, A. Beygelzimer, M. Dudfk, J. Langford, and W. Hanna. A reductions approach to fair classification. In *35th International Conference on Machine Learning, ICML 2018*, 2018. ISBN 9781510867963.

R. Agarwal, L. Melnick, N. Frosst, X. Zhang, B. Lengerich, R. Caruana, and G. Hinton. Neural Additive Models: Interpretable Machine Learning with Neural Nets, 2021.

G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes, 2018.

R. Andrews, J. Diederich, and A. B. Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 1995. ISSN 09507051. doi: 10.1016/0950-7051(96)81920-4.

I. Arel, D. Rose, and T. Karnowski. Deep machine learning-A new frontier in artificial intelligence research. *IEEE Computational Intelligence Magazine*, 2010. ISSN 1556603X. doi: 10.1109/MCI.2010.938364.

L. Arras, F. Horn, G. Montavon, K.-R. Müller, and W. Samek. Explaining Predictions of Non-Linear Classifiers in NLP. 2016. doi: 10.18653/v1/w16-1601.

V. Arya, R. K. E. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S. C. Hoffman, S. Houde, Q. V. Liao, R. Luss, A. Mojsilović, S. Mourad, P. Pedemonte, R. Raghavendra, J. Richards, P. Sattigeri, K. Shanmugam, M. Singh, K. R. Varshney, D. Wei, and Y. Zhang. One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques, 2019.

S. Bader and P. Hitzler. Dimensions of Neural-symbolic Integration - A Structured Survey. 11 2005. URL `http://arxiv.org/abs/cs/0511042`.

S. Badreddine, A. d. Garcez, L. Serafini, and M. Spranger. Logic Tensor Networks. 2020. URL http://arxiv.org/abs/2012.13635.

D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. ISBN 9781538604571. doi: 10.1109/CVPR.2017.354.

F. Bianchi and P. Hitzler. On the Capabilities of Logic Tensor Networks for Deductive Reasoning. *AAAI Spring Symposium MAKE*, 2019.

J. Bien and R. Tibshirani. Prototype selection for interpretable classification. *Annals of Applied Statistics*, 2011. ISSN 19326157. doi: 10.1214/11-AOAS495.

L. Bottou. From machine learning to machine reasoning: An essay. *Machine Learning*, 2014. ISSN 08856125. doi: 10.1007/s10994-013-5335-x.

R. Brachman and H. Levesque. *Knowledge Representation and Reasoning*, volume 1. 2004. ISBN 1558609326. doi: 10.1146/annurev.cs.01.060186.001351. URL http://books.google.be/books?id=OuPtLaA5QjoC.

J. Burrell. How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big Data & Society*, 2016. ISSN 2053-9517. doi: 10.1177/2053951715622512.

A. Cangelosi. The grounding and sharing of symbols. *Pragmatics & Cognition*, 2006. ISSN 0929-0907. doi: 10.1075/pc.14.2.08can.

R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission. In *21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15), Proceedings*, 2015. ISBN 9781450336642. doi: 10.1145/2783258.2788613.

D. V. Carvalho, E. M. Pereira, and J. S. Cardoso. Machine learning interpretability: A survey on methods and metrics, 2019. ISSN 20799292.

Z. Che, S. Purushotham, R. Khemani, and Y. Liu. Interpretable Deep Models for ICU Outcome Prediction. *AMIA Annual Symposium Proceedings*, 2016. ISSN 1942-597X.

Y. Choi, G. Farnadi, B. Babaki, and G. V. d. Broeck. Learning Fair Naive Bayes Classifiers by Discovering and Eliminating Discrimination Patterns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019. URL http://arxiv.org/abs/1906.03843.

F. Chollet. Keras. *GitHub Repository*, 2015. URL https://github.com/keras-team/keras.

R. Confalonieri, T. Weyde, T. R. Besold, and F. Moscoso del Prado Martín. Using ontologies to enhance human understandability of global post-hoc explanations of black-box models. *Artificial Intelligence*, 2021. ISSN 00043702. doi: 10.1016/j.artint.2021.103471.

M. W. Craven. *Extracting Comprehensible Models From Trained Neural Networks.* PhD thesis, University of Wisconsin, Madison, 1996.

A. Daniele and L. Serafini. Knowledge enhanced neural networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019. ISBN 9783030299071. doi: 10.1007/978-3-030-29908-8_43.

A. Dhurandhar, R. Luss, K. Shanmugam, and P. Olsen. Improving simple models with confidence profiles. In *Advances in Neural Information Processing Systems*, 2018.

M. Diligenti, M. Gori, and C. Saccà. Semantic-based regularization for learning and inference. *Artificial Intelligence*, 2017. ISSN 00043702. doi: 10.1016/j.artint.2015.08.011.

I. Donadello, L. Serafini, and A. d. Garcez. Logic tensor networks for semantic image interpretation. In *IJCAI International Joint Conference on Artificial Intelligence*, 2017. ISBN 9780999241103. doi: 10.24963/ijcai.2017/221.

D. Doran, S. Schulz, and T. R. Besold. What does explainable AI really mean? A new conceptualization of perspectives. In *CEUR Workshop Proceedings*, 2018.

J. Drake. Explanatory Power. In *Introduction to Logic*, pages 160–161. EP TECH PRESS, 2018. ISBN 978-1-83947-421-7.

C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *ITCS 2012 - Innovations in Theoretical Computer Science Conference*, 2012. ISBN 9781450311151. doi: 10.1145/2090236.2090255.

M. Ebrahimi, A. Eberhart, F. Bianchi, and P. Hitzler. Towards bridging the neuro-symbolic gap: deep deductive reasoners. *Applied Intelligence*, 2021. ISSN 15737497. doi: 10.1007/s10489-020-02165-6.

European Union. Regulation 2016/679 of the European parliament and the Council of the European Union. *Official Journal of the European Communities*, 2016.

R. Fagin, R. Riegel, and A. Gray. Foundations of Reasoning with Uncertainty via Real-valued Logics, 2021.

M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015. ISBN 9781450336642. doi: 10.1145/2783258.2783311.

M. V. França, G. Zaverucha, and A. d. Garcez. Fast relational learning using bottom clause propositionalization with artificial neural networks. *Machine Learning*, 2014. ISSN 08856125. doi: 10.1007/s10994-013-5392-1.

R. M. French. Catastrophic forgetting in connectionist networks, 1999. ISSN 13646613.

S. A. Friedler, S. Choudhary, C. Scheidegger, E. P. Hamilton, S. Venkatasubramanian, and D. Roth. A comparative study of fairness-enhancing interventions in machine learning. In *FAT* 2019 - Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency*, 2019. ISBN 9781450361255. doi: 10.1145/3287560.3287589.

J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 2001. ISSN 00905364.

N. Frosst and G. Hinton. Distilling a Neural Network Into a Soft Decision Tree. 2017. URL `http://arxiv.org/abs/1711.09784`.

K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 1980. ISSN 03401200. doi: 10.1007/BF00344251.

G. Futia and A. Vetrò. On the integration of knowledge graphs into deep learning models for a more comprehensible AI-Three challenges for future research, 2020. ISSN 20782489.

A. d. Garcez, K. Broda, and D. M. Gabbay. Symbolic knowledge extraction from trained neural networks: A sound approach. *Artificial Intelligence*, 2001. ISSN 00043702. doi: 10.1016/S0004-3702(00)00077-1.

A. d. Garcez, K. B. Broda, and D. M. Gabbay. *Neural-Symbolic Learning Systems.* Perspectives in Neural Computing. Springer London, London, 2002. ISBN 978-1-85233-512-0. doi: 10.1007/978-1-4471-0211-3. URL `http://link.springer.com/10.1007/978-1-4471-0211-3`.

A. d. Garcez, L. C. Lamb, and D. M. Gabbay. *Neural-symbolic cognitive reasoning.* Springer Science & Business Media, 2008. ISBN 3540732462.

A. d. Garcez, M. Gori, L. C. Lamb, L. Serafini, M. Spranger, and S. N. Tran. Neural-Symbolic Computing: An Effective Methodology for Principled Integration of Machine Learning and Reasoning. *arXiv:1905.06088*, 2019. URL `http://arxiv.org/abs/1905.06088`.

M. Garnelo and M. Shanahan. Reconciling deep learning with symbolic artificial intelligence: representing objects and relations, 2019. ISSN 23521546.

A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin. Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics*, 2015. ISSN 15372715. doi: 10.1080/10618600.2014.907095.

GoogleDevelopers. Embeddings: Translating to a Lower-Dimensional Space, 2021.

M. Graziani, V. Andrearczyk, and H. Müller. Regression Concept Vectors for Bidirectional Explanations in Histopathology. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 124–132. 2018. ISBN 9783030026271.

R. Guidotti, A. Monreale, F. Turini, D. Pedreschi, and F. Giannotti. A Survey Of Methods For Explaining Black Box Models. *ACM Computing Surveys*, pages 1–45, 2018. URL `https://doi.org/10.1145/3236009`.

S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo. Jointly embedding knowledge graphs and logical rules. In *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2016. ISBN 9781945626258. doi: 10.18653/v1/d16-1019.

R. S. Gutzwiller and J. Reeder. Dancing With Algorithms: Interaction Creates Greater Preference and Trust in Machine-Learned Behavior. *Human Factors*, 2021. ISSN 15478181. doi: 10.1177/0018720820903893.

S. Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 1990. ISSN 01672789. doi: 10.1016/0167-2789(90)90087-6.

C. G. Hempel. I.—STUDIES IN THE LOGIC OF CONFIRMATION (I.). *Mind*, 1945. ISSN 0026-4423. doi: 10.1093/mind/liv.213.1.

G. Hinton, O. Vinyals, and J. Dean. Dark knowledge. *Presented as the keynote in BayLearn*, 2, 2014.

S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 1997. ISSN 08997667. doi: 10.1162/neco.1997.9.8.1735.

G. Hoffman. How neural networks learn distributed representations. *Oreilly Media*, 2018.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 1989. ISSN 08936080. doi: 10.1016/0893-6080(89)90020-8.

House of Lords and Select Committee on Artificial Intelligence. *AI in the UK: Ready, Willing and Able? Report of Session 2017-19.* Number v. 1. Authority of the House of Lords, 2018.

Z. Hu, X. Ma, Z. Liu, E. Hovy, and E. P. Xing. Harnessing deep neural networks with logic rules. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, 2016. ISBN 9781510827585.

I. T. Jolliffe. Principal Component Analysis, Second Edition. *Encyclopedia of Statistics in Behavioral Science*, 2002. ISSN 00401706. doi: 10.2307/1270093.

M. I. Jordan and R. A. Jacobs. Hierarchical Mixtures of Experts and the EM Algorithm. *Neural Computation*, 6(2):181–214, 1994. ISSN 0899-7667. doi: 10.1162/neco.1994.6.2.181. URL `http://www.mitpressjournals.org/doi/10.1162/neco.1994.6.2.181`.

W. Jue and P. Domingos. Hybrid markov logic networks. In *Proceedings of the National Conference on Artificial Intelligence*, 2008. ISBN 9781577353683.

L. Kaufman and P. J. Rousseeuw. Partitioning Around Medoids (Program PAM), in Finding Groups in Data: An Introduction to Cluster Analysis. *Wiley, Hoboken*, 2008.

B. Khaleghi. An Explanation of What, Why, and How of eXplainable AI (XAI), 2019. URL `https://www.youtube.com/watch?v=rPSiEDYcXr4`.

S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah. Transformers in Vision: A Survey, 2021.

M. Khodak, N. Saunshi, and K. Vodrahalli. A large self-annotated corpus for sarcasm. In *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, 2019. ISBN 9791095546009.

B. Kim, C. Rudin, and J. Shah. The Bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Advances in Neural Information Processing Systems*, 2014.

B. Kim, R. Khanna, and O. Koyejo. Examples are not Enough, Learn to Criticize! Criticism for Interpretability. In *Neural Information Processing Systems*, 2016. ISBN 9783642327223. doi: 10.1371/journal.pone.0034113.

B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). *arXiv:1711.11279*, 2017. URL `http://arxiv.org/abs/1711.11279`.

B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. Interpretability beyond feature attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *35th International Conference on Machine Learning, ICML 2018*, 2018. ISBN 9781510867963.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

J. Kleinberg, J. Ludwig, S. Mullainathan, and A. Rambachan. Algorithmic Fairness. *AEA Papers and Proceedings*, 2018. ISSN 2574-0768. doi: 10.1257/pandp.20181018.

T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 1982. ISSN 03401200. doi: 10.1007/BF00337288.

E. v. Krieken, E. Acar, and F. v. Harmelen. Analyzing Differentiable Fuzzy Logic Operators. 2020. URL `https://arxiv.org/abs/2002.06100v1`.

S. Krishnan and E. Wu. PALM: Machine Learning Explanations For Iterative Debugging. *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics - HILDA'17*, 2017. doi: 10.1145/3077257.3077271.

A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. *... Science Department, University of Toronto, Tech. ...*, 2009. ISSN 1098-6596. doi: 10.1.1.222.9220.

A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial machine learning at scale. In *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017.

H. Lakkaraju, S. H. Bach, and J. Leskovec. Interpretable decision sets: a joint framework for description and prediction. *International Conference on Knowledge Discovery and Data Mining*, 2016. ISSN 2154-817X. doi: 10.1145/2939672.2939874.

T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse Processes*, 1998. ISSN 0163-853X. doi: 10.1080/01638539809545028.

J. Larson, S. Mattu, A. Kirchner, and J. Angwin. How We Analyzed the COMPAS Recidivism Algorithm. *ProPublica*, 2016.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

Y. LeCun, K. Weinberger, P. Simard, and R. Caruana. Panel debate on Interpretability, 2017. URL `https://www.youtube.com/watch?v=2hW05ZfsUUo`.

LeCun Yann, Cortes Corinna, and Burges Christopher. THE MNIST DATABASE of handwritten digits. *The Courant Institute of Mathematical Sciences*, 1998.

B. Letham, C. Rudin, T. H. McCormick, and D. Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *Annals of Applied Statistics*, 2015. ISSN 19417330. doi: 10.1214/15-AOAS848.

S. W. Littlejohn. Symbolic interactionism as an approach to the study of human communication. *Quarterly Journal of Speech*, 1977. ISSN 0033-5630. doi: 10.1080/00335637709383369.

S. M. Lundberg and S. I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, 2017.

D. M. Malioutov, K. R. Varshney, A. Emad, and S. Dash. Learning Interpretable Classification Rules with Boolean Compressed Sensing. 2017. doi: 10.1007/978-3-319-54024-5_5.

R. Manhaeve, A. Kimmig, S. Dumančić, T. Demeester, and L. De Raedt. Deepproblog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems*, 2018.

G. Marra, F. Giannini, M. Diligenti, and M. Gori. LYRICS: a General Interface Layer to Integrate Logic Inference and Deep Learning. 2019. URL `http://arxiv.org/abs/1903.07534`.

G. Marra, F. Giannini, M. Diligenti, and M. Gori. Integrating Learning and Reasoning with Deep Logic Models. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020. ISBN 9783030461461. doi: 10.1007/978-3-030-46147-8_31.

G. Marra, S. Dumančić, R. Manhaeve, and L. De Raedt. From Statistical Relational to Neural Symbolic Artificial Intelligence: a Survey. 8 2021. URL `http://arxiv.org/abs/2108.11451`.

J. McCarthy. Programs with common sense. *Proceedings of the Symposium on the Mechanization of Thought Processes*, 1963.

W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 1943. ISSN 00074985. doi: 10.1007/BF02478259.

N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A Survey on Bias and Fairness in Machine Learning. 2019. URL `http://arxiv.org/abs/`.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations ofwords and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 2013.

T. Miller. Explanation in Artificial Intelligence: Insights from the Social Sciences. *arXiv:1706.07269*, 2017. doi: arXiv:1706.07269v1.

C. Molnar. *Interpretable Machine Learning*. Christoph Molnar, 2019. URL `https://christophm.github.io/interpretable-ml-book/`.

G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K. R. Müller. Layer-Wise Relevance Propagation: An Overview. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2019. doi: 10.1007/978-3-030-28954-6_10.

M. Mundt, Y. W. Hong, I. Pliushch, and V. Ramesh. A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning, 2020. ISSN 23318422.

W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv:1901.04592*, 2019. URL `http://arxiv.org/abs/1901.04592`.

A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *arXiv:1605.09304*, 2016. URL `http://arxiv.org/abs/1605.09304`.

D. T. Nguyen, K. E. Kasmarik, and H. A. Abbass. Towards Interpretable ANNs: An Exact Transformation to Multi-Class Multivariate Decision Trees, 2021.

W. Nie, Y. Zhang, and A. Patel. A Theoretical Explanation for Perplexing Behaviors of Backpropagation-based Visualizations. *arXiv:1805.07039*, 2018. URL `http://arxiv.org/abs/1805.07039`.

S. Odense. *Layerwise symbolic knowledge extraction from deep neural networks.* PhD thesis, City, University of London, 2019. URL `https://openaccess.city.ac.uk/id/eprint/24700/`.

S. Odense and A. d'Avila Garcez. Layerwise Knowledge Extraction from Deep Convolutional Networks. *CoRR*, abs/2003.0, 2020. URL `https://arxiv.org/abs/2003.09000`.

M. Padala and S. Gujar. FNNC: Achieving Fairness through Neural Networks. In C. Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, {IJCAI-20}*. International Joint Conferences on Artificial Intelligence Organization, 2020.

G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review, 2019. ISSN 18792782.

K. Pearson. LIII. On lines and planes of closest fit to systems of points in space . *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1901. ISSN 1941-5982. doi: 10.1080/14786440109462720.

S. T. Piantadosi. The Computational Origin of Representation. *Minds and Machines*, 2021. ISSN 15728641. doi: 10.1007/s11023-020-09540-9.

G. Piccinini and S. Bahar. Neural Computation and the Computational Theory of Cognition. *Cognitive Science*, 2013. ISSN 03640213. doi: 10.1111/cogs.12012.

G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger. On fairness and calibration. In *Advances in Neural Information Processing Systems*, 2017.

A. Preece, D. Harborne, D. Braines, R. Tomsett, and S. Chakraborty. Stakeholders in Explainable AI. *arXiv:1810.00184*, 2018. URL `http://arxiv.org/abs/1810.00184`.

Y. Qi, D. S. Sachan, M. Felix, S. J. Padmanabhan, and G. Neubig. When and why are pre-trainedword embeddings useful for neural machine translation? In *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2018. ISBN 9781948087292. doi: 10.18653/v1/n18-2084.

J. R. Quinlan. *C4.5: Programs for Machine Learning*, volume 1. 1992. ISBN 1558602380. doi: 10.1016/S0019-9958(62)90649-6.

A. Radford, J. Wook, K. Chris, H. Aditya, R. Gabriel, G. Sandhini, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning Transferable Visual Models From Natural Language Supervision. 2021.

M. T. Ribeiro, S. Singh, and C. Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *arXiv:1602.04938*, 2016. URL `http://arxiv.org/abs/1602.04938`.

M. Richardson and P. Domingos. Markov logic networks. In *Machine Learning*, 2006. doi: 10.1007/s10994-006-5833-1.

R. Riegel, A. Gray, F. Luus, N. Khan, N. Makondo, I. Y. Akhalwaya, H. Qian, R. Fagin, F. Barahona, U. Sharma, S. Ikbal, H. Karanam, S. Neelam, A. Likhyani, and S. Srivastava. Logical Neural Networks, 2020.

F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 1958. doi: 10.1037/h0042519.

C. Rudin. Please Stop Explaining Black Box Models for High Stakes Decisions. *arXiv:1811.10154*, 2018. URL `http://arxiv.org/abs/1811.10154`.

C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 2019. doi: 10.1038/s42256-019-0048-x.

S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* 2010. ISBN 0137903952. doi: 10.1017/S0269888900007724.

S. Sarkar, T. Weyde, A. d. Garcez, G. Slabaugh, S. Dragicevic, and C. Percy. Accuracy and Interpretability Trade-offs in Machine Learning Applied to Safer Gambling. *CEUR Workshop Proceedings*, (1773), 2016. ISSN 1201-. URL `http://openaccess.city.ac.uk/16484/`.

R. Schwarzenberg, M. d. Hu, D. Harbecke, C. Alt, and L. Hennig. Layerwise relevance visualization in convolutional text graph classifiers. In *EMNLP-IJCNLP 2019 - Graph-Based Methods for Natural Language Processing - Proceedings of the 13th Workshop*, 2019. ISBN 9781950737864. doi: 10.18653/v1/d19-5308.

L. Serafini and A. d. Garcez. Logic tensor networks: Deep learning and logical reasoning from data and knowledge. *arXiv preprint arXiv:1606.04422*, 2016.

L. S. Shapley. Contributions to the Theory of Games. In *Annals of Mathematical Studies v. 28*. 1953.

H. Singh, M. Aggarwal, and B. Krishnamurthy. Exploring Neural Models for Parsing Natural Language into First-Order Logic. *CoRR*, abs/2002.06544, 2020. URL `https://arxiv.org/abs/2002.06544`.

R. Socher, D. Chen, C. D. Manning, and A. Y. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, 2013.

R. Speer, J. Chin, and C. Havasi. {ConceptNet} 5.5: An Open Multilingual Graph of General Knowledge. pages 4444–4451, 2017. URL `http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972`.

E. Štrumbelj and I. Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 2014. ISSN 02193116. doi: 10.1007/s10115-013-0679-x.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015. ISBN 9781467369640. doi: 10.1109/CVPR.2015.7298594.

S. Tan, R. Caruana, G. Hooker, and A. Gordo. Transparent Model Distillation. 2018a. URL `http://arxiv.org/abs/1801.08640`.

S. Tan, R. Caruana, G. Hooker, and Y. Lou. Distill-and-Compare: Auditing Black-Box Models Using Transparent Model Distillation. In *AIES 2018 - Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018b. ISBN 9781450360128. doi: 10.1145/3278721.3278725.

J. Townsend, T. Kasioumis, and H. Inakoshi. ERIC: Extracting Relations Inferred from Convolutions. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2021. ISBN 9783030695347. doi: 10.1007/978-3-030-69535-4_13.

B. Ustun and C. Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 2016. ISSN 15730565. doi: 10.1007/s10994-015-5528-6.

L. Van Der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008. ISSN 15324435.

L. J. P. Van Der Maaten, E. O. Postma, and H. J. Van Den Herik. Dimensionality Reduction: A Comparative Review. *Journal of Machine Learning Research*, 2009. ISSN 0169328X. doi: 10.1080/13506280444000102.

E. van Krieken, E. Acar, and F. van Harmelen. Semi-Supervised Learning using Differentiable Reasoning. *Journal of Applied Logics — IfCoLog Journal of Logics and their Applications*, 6(4), 8 2019. URL `http://arxiv.org/abs/1908.04700`.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. *SSRN*, 2017. ISSN 1556-5068. doi: 10.2139/ssrn.3063289.

A. White and A. d. Garcez. Measurable Counterfactual Local Explanations for Any Classifier. In *24th European Conference on Artificial Intelligence*, 2020. URL `http://arxiv.org/abs/1908.03020`.

M. Wu, M. C. Hughes, S. Parbhoo, M. Zazzi, V. Roth, and F. Doshi-Velez. Beyond sparsity: Tree regularization of deep models for interpretability. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv:1708.07747*, 2017. URL `http://arxiv.org/abs/1708.07747`.

K. Xu, D. H. Park, C. Yi, and C. Sutton. Interpreting Deep Classifier by Visual Distillation of Dark Knowledge. *arXiv preprint arXiv:1803.04042*, 2018.

C. Yang, A. Rangarajan, and S. Ranka. Global Model Interpretation via Recursive Partitioning. *arXiv:1802.04253*, 2018. URL `http://arxiv.org/abs/1802.04253`.

J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, 2014.

O. T. Yıldız, E. Alpaydin, and O. Irsoy. Soft Decision Trees. *International Conference on Pattern Recognition*, 1(Icpr):1819–1822, 2012.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014. ISBN 9783319105895. doi: 10.1007/978-3-319-10590-1_53.

R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. In *30th International Conference on Machine Learning, ICML 2013*, 2013.