



City Research Online

City, University of London Institutional Repository

Citation: Garcez, A., Gori, M., Lamb, L. C., Serafini, L., Spranger, M. and Tran, S. N. (2019). Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *Journal of Applied Logics*, 6(4), pp. 611-632.

This is the accepted version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: <http://openaccess.city.ac.uk/id/eprint/22984/>

Link to published version:

Copyright and reuse: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

City Research Online:

<http://openaccess.city.ac.uk/>

publications@city.ac.uk

NEURAL-SYMBOLIC COMPUTING: AN EFFECTIVE METHODOLOGY FOR PRINCIPLED INTEGRATION OF MACHINE LEARNING AND REASONING

ARTUR D'AVILA GARCEZ, MARCO GORI, LUIS C. LAMB, LUCIANO SERAFINI,
MICHAEL SPRANGER, SON N. TRAN*

*City, Univ. of London, Univ. of Siena, UFRGS, FBK, Sony Japan, Univ. of
Tasmania*

a.garcez@city.ac.uk, mxgori@gmail.com, luislamb@acm.org,
serafini@fbk.eu, michael.spranger@gmail.com, sn.tran@utas.edu.au

Abstract

Current advances in Artificial Intelligence and machine learning in general, and deep learning in particular have reached unprecedented impact not only across research communities, but also over popular media channels. However, concerns about interpretability and accountability of AI have been raised by influential thinkers. In spite of the recent impact of AI, several works have identified the need for principled knowledge representation and reasoning mechanisms integrated with deep learning-based systems to provide sound and explainable models for such systems. Neural-symbolic computing aims at integrating, as foreseen by Valiant, two most fundamental cognitive abilities: the ability to learn from the environment, and the ability to reason from what has been learned. Neural-symbolic computing has been an active topic of research for many years, reconciling the advantages of robust learning in neural networks and reasoning and interpretability of symbolic representation. In this paper, we survey recent accomplishments of neural-symbolic computing as a principled methodology for integrated machine learning and reasoning. We illustrate the effectiveness of the approach by outlining the main characteristics of the methodology: principled integration of neural learning with symbolic knowledge representation and reasoning allowing for the construction of explainable AI systems. The insights provided by neural-symbolic computing shed new light on the increasingly prominent need for interpretable and accountable AI systems.

We thank Richard Evans for his valuable comments and suggestions.

*Corresponding author. Authors are in alphabetical order.

1 Introduction

Current advances in Artificial Intelligence (AI) and machine learning in general, and deep learning in particular have reached unprecedented impact not only within the academic and industrial research communities, but also among popular media channels. Deep learning researchers have achieved groundbreaking results and built AI systems that have in effect rendered new paradigms in areas such as computer vision, game playing, and natural language processing [25, 43]. Nonetheless, the impact of deep learning has been so remarkable that leading entrepreneurs such as Elon Musk and Bill Gates, and outstanding scientists such as Stephen Hawking have voiced strong concerns about AI’s accountability, impact on humanity and even on the future of the planet [38].

Against this backdrop, researchers have recognised the need for offering a better understanding of the underlying principles of AI systems, in particular those based on machine learning, aiming at establishing solid foundations for the field. In this respect, Turing Award Winner Leslie Valiant had already pointed out that one of the key challenges for AI in the coming decades is the development of integrated reasoning and learning mechanisms, so as to construct a rich semantics of intelligent cognitive behavior [51]. In Valiant’s words: *“The aim here is to identify a way of looking at and manipulating commonsense knowledge that is consistent with and can support what we consider to be the two most fundamental aspects of intelligent cognitive behavior: the ability to learn from experience, and the ability to reason from what has been learned. We are therefore seeking a semantics of knowledge that can computationally support the basic phenomena of intelligent behavior.”* In order to respond to these scientific, technological and societal challenges which demand reliable, accountable and explainable AI systems and tools, the integration of cognitive abilities ought to be carried out in a principled way.

Neural-symbolic computing aims at integrating, as put forward by Valiant, two most fundamental cognitive abilities: the ability to learn from experience, and the ability to reason from what has been learned [1, 11, 15].

The integration of learning and reasoning through neural-symbolic computing has been an active branch of AI research for several years [13, 15, 16, 20, 23, 40, 50]. Neural-symbolic computing aims at reconciling the dominating symbolic and connectionist paradigms of AI under a principled foundation. In neural-symbolic computing, knowledge is represented in symbolic form, whereas learning and reasoning are computed by a neural network. Thus, the underlying characteristics of neural-symbolic computing allow the principled combination of robust learning and efficient inference in neural networks, along with interpretability offered by symbolic knowledge extraction and reasoning with logical systems.

Importantly, as AI systems started to outperform humans in certain tasks [43], several ethical and societal concerns were raised [38]. Therefore, the interpretability and explainability of AI systems become crucial alongside their accountability.

In this paper, we survey the principles of neural-symbolic integration by highlighting key characteristics that underline this research paradigm. Despite their differences, both the symbolic and connectionist paradigms, share common characteristics offering benefits when integrated in a principled way (see e.g. [7, 15, 44, 50]). For instance, neural learning and inference under uncertainty may address the brittleness of symbolic systems. On the other hand, symbolism provides additional knowledge for learning which may e.g. ameliorate neural network’s well-known catastrophic forgetting or difficulty with extrapolating. In addition, the integration of neural models with logic-based symbolic models provides an AI system capable of bridging lower-level information processing (for perception and pattern recognition) and higher-level abstract knowledge (for reasoning and explanation).

In what follows, we review the important and recent developments of research on neural-symbolic systems. We start by outlining the main important characteristics of a neural-symbolic system: Representation, Extraction, Reasoning and Learning [1, 16], and their applications. We then discuss and categorise the approaches to representing symbolic knowledge in neural-symbolic systems into three main groups: rule-based, formula-based and embedding-based. After that, we show the capabilities and applications of neural-symbolic systems for learning, reasoning, and explainability. Towards the end of the paper we outline recent trends and identify a few challenges for neural-symbolic computing research.

2 Prolegomenon to Neural-Symbolic Computing

Neural-symbolic systems have been applied successfully to several fields, including data science, ontology learning, training and assessment in simulators, and models of cognitive learning and reasoning [4, 13, 15, 32]. However, the recent impact of deep learning in vision and language processing and the growing complexity of (autonomous) AI systems demand improved explainability and accountability. In neural-symbolic computing, learning, reasoning and knowledge extraction are combined. Neural-symbolic systems are modular and seek to have the property of compositionality. This is achieved through the streamlined representation of several knowledge representation languages which are computed by connectionist models. The *Knowledge-Based Artificial Neural Network* (KBANN) [47] and the *Connectionist inductive learning and logic programming* (CILP) [16] systems were some of the most influential models that combine logical reasoning and neural learning. As

pointed out in [16] KBANN served as inspiration in the construction of the CILP system. CILP provides a sound theoretical foundation to inductive learning and reasoning in artificial neural networks through theorems showing how logic programming can be a knowledge representation language for neural networks. The KBANN system was the first to allow for learning with background knowledge in neural networks and knowledge extraction, with relevant applications in bioinformatics. CILP allowed for the integration of learning, reasoning and knowledge extraction in recurrent networks. An important result of CILP was to show how neural networks endowed with semi-linear neurons approximate the fixed-point operator of propositional logic programs with negation. This result allowed applications of reasoning and learning using backpropagation and logic programs as background knowledge [16].

Notwithstanding, the need for richer cognitive models soon demanded the representation and learning of other forms of reasoning, such as temporal reasoning, reasoning about uncertainty, epistemic, constructive and argumentative reasoning [15, 51]. Modal and temporal logic have achieved first class status in the formal toolboxes of AI and Computer Science researchers. In AI, modal logics are amongst the most widely used logics in the analysis and modelling of reasoning in distributed multiagent systems. In the early 2000s, researchers then showed that ensembles of CILP neural networks, when properly set up, can compute the *modal* fixed-point operator of modal and temporal logic programs. In addition to these results, such ensembles of neural networks were shown to represent the possible world semantics of modal propositional logic, fragments of first order logic and of linear temporal logics. In order to illustrate the computational power of *Connectionist Modal Logics (CML)* and *Connectionist Temporal Logics of Knowledge (CTLK)* [7, 8], researchers were able to learn full solutions to several problems in distributed, multiagent learning and reasoning, including the Muddy Children Puzzle [7] and the Dining Philosophers Problem [24].

By combining temporal logic with modalities, one can represent knowledge and learning evolution in time. This is a key insight, allowing for temporal evolution of both learning and reasoning in time (see Fig. 1). The Figure represents the integrated learning and reasoning process of CTLK. At each time point (or one state of affairs), e.g. t_2 , knowledge which the agents are endowed with and what the agents have learned at the previous time t_1 is represented. As time progresses, linear evolution of the agents' knowledge is represented in time as more knowledge about the world (what has been learned) is represented. Fig. 1 illustrates this dynamic property of CTLK, which allows not only the analysis of the current state of affairs but also of how knowledge and learning evolve over time.

Modal and temporal reasoning, when integrated with connectionist learning pro-

vide neural-symbolic systems with richer knowledge representation languages and better interpretability. As can be seen in Fig. 1, they enable the construction of more modular deep networks. As argued by Valiant, the construction of cognitive models integrating rich logic-based knowledge representation languages, with robust learning algorithms provide an effective alternative to the construction of semantically sound cognitive neural computational models. It is also argued that a language for describing the algorithms of deep neural networks is needed. Non-classical logics such as logic programming in the context of neuro-symbolic systems, and functional languages used in the context of probabilistic programming are two prominent candidates. In the coming sections, we explain how neural-symbolic systems can be constructed from simple definitions which underline the streamlined integration of knowledge representation, learning, and reasoning in a unified model.

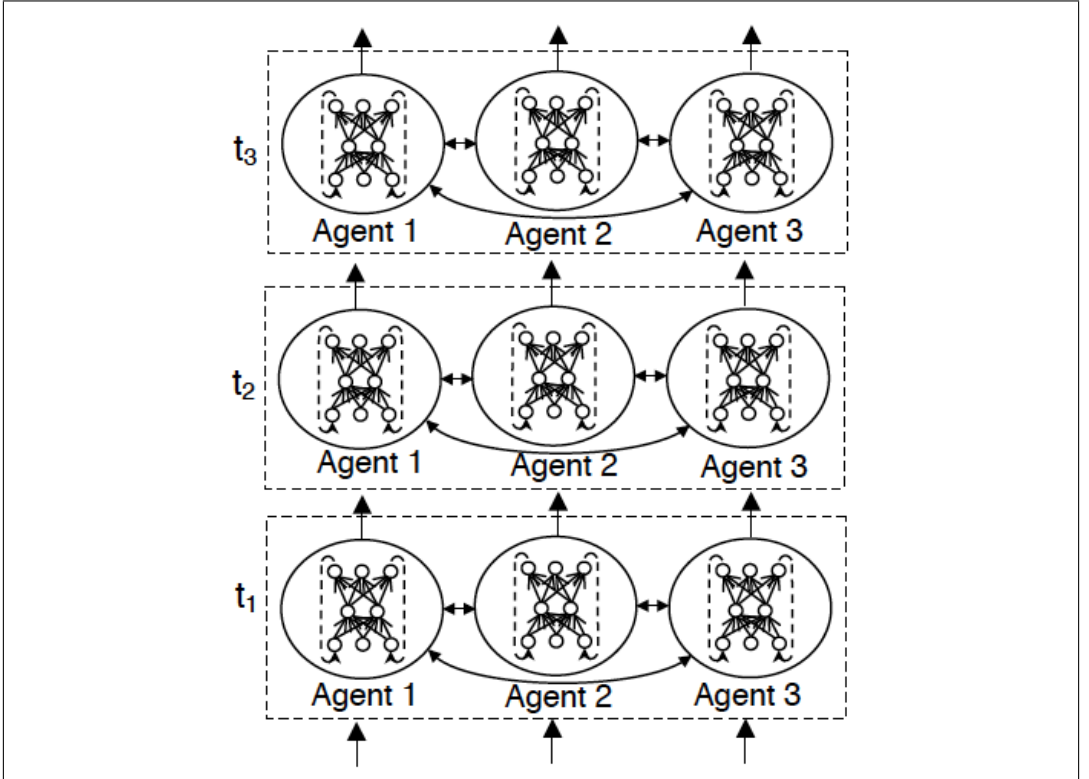


Figure 1: Evolution of Reasoning and Learning in Time

3 Knowledge Representation in Neural Networks

Knowledge representation is the cornerstone of a neural-symbolic system that provides a mapping mechanism between symbolism and connectionism, where logical calculus can be carried out exactly or approximately by a neural network. This way, given a trained neural network, symbolic knowledge can be extracted for explaining and reasoning purposes. The representation approaches can be categorised into three main groups: rule-based, formula-based and embedding, which are discussed as follows.

3.1 Propositional Logic

3.1.1 Rule-based Representation

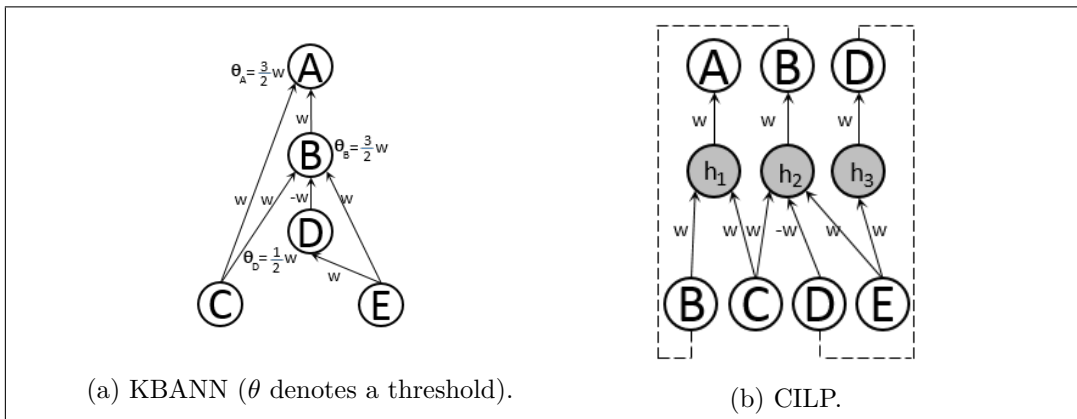


Figure 2: Knowledge representation of $\phi = \{A \leftarrow B \wedge C, B \leftarrow C \wedge \neg D \wedge E, D \leftarrow E\}$ using KBANN and CILP.

Early work on representation of symbolic knowledge in connectionist networks focused on tailoring the models' parameters to establish an equivalence between input-output mapping function of artificial neural networks (ANN) and logical inference rules. It has been shown that by constraining the weights of a neural network, inference with feedforward propagation can exactly imitate the behaviour of modus ponens [47, 6]. KBANN [47] employs stack of perceptrons to represent the inference rule of logical implications. For example, given a set of rules:

$$\phi = \{A \leftarrow B \wedge C, B \leftarrow C \wedge \neg D \wedge E, D \leftarrow E\} \quad (1)$$

an ANN can be constructed as in Figure 2a. CILP then generalises the idea by using recurrent networks and bounded continuous units [6]. This representation

method allows the use of various data types and more complex sets of rules. With CILP, knowledge given in Eq. (1) can be encoded in a neural network as shown in Figure 2b. In order to adapt this system to first-order logic, CILP++ [14] makes use of techniques from Inductive Logic Programming (ILP). In CILP++, examples and background knowledge are converted into propositional clauses by a bottom-clause propositionalisation technique, which are then encoded into an ANN with recurrent connections as done by CILP.

3.1.2 Formula-based Representation

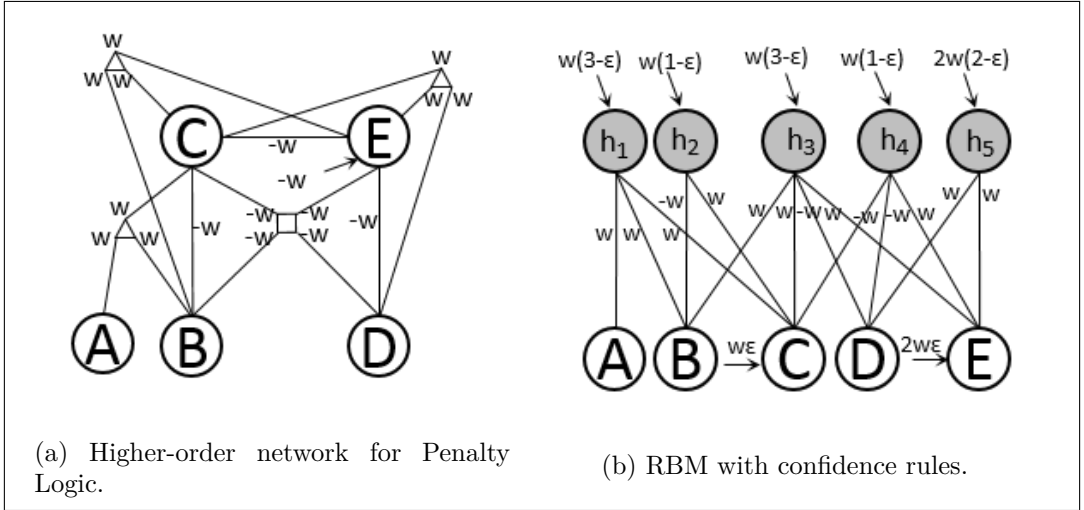


Figure 3: Knowledge representation of $\phi = \{w : A \leftarrow B \wedge C, w : B \leftarrow C \wedge \neg D \wedge E, w : D \leftarrow E\}$ using Penalty logic and Confidence rules

One issue with KBANN-style rule-based representations is that the discriminative structure of ANNs will only allow a subset of the variables (the consequent of the if-then formula) to be inferred, unless recurrent networks are deployed, with the other variables (the antecedents) being seen as inputs only. This would not represent the behaviour of logical formulas and does not support general reasoning where any variable can be inferred. In order to solve this issue, generative neural networks can be employed as they can treat all variables as non-discriminative. In this formula-based approach, typically associated with restricted Boltzmann machines (RBMs) as a building block, the focus is on mapping logical formulas to symmetric connectionist networks, each characterised by an energy function. Early work such as penalty logic [33] proposes a mechanism to represent weighted formulas in energy-

based connectionist (Hopfield) networks where maximising satisfiability is equivalent to minimising energy function. Suppose that each formula in the knowledge base (1) is assigned a weight w . Penalty logic constructs a higher-order Hopfield network as shown in Figure 3a. However, inference with such type of network is difficult, while converting the higher-order energy function to a quadratic form is possible but computationally expensive. Recent work on confidence rules [48] proposes an efficient method to represent propositional formulas in restricted Boltzmann machines and deep belief networks where inference and learning become easier. Figure 3b shows an RBM for the knowledge base (1). Nevertheless, learning and reasoning with restricted Boltzmann machines are still complex, making it more difficult to apply formula-based representations than rule-based representations in practice. The main issue has to do with the partition functions of symmetric connectionist networks which cannot be computed analytically. This intractability problem, fortunately, can be ameliorated using sum-product approach as has been shown in [36]. However, it is not yet clear how to apply this idea to RBMs.

3.2 First-order Logic

3.2.1 Propositionalisation

Representation of knowledge in first-order logic in neural networks has been an on-going challenge, but it can benefit from studies of propositional logic representation 3.1 using propositionalisation techniques [28]. Such techniques allow a first-order knowledge base to be converted into a propositional knowledge base so as to preserve entailment. In neural-symbolic computing, bottom clause propositionalisation (BCP) is a popular approach because bottom clause literals can be encoded directly into neural networks as data features while at the same time presenting semantic meaning.

Early work from [10] employs propositionalisation and feedforward neural networks to learn a clause evaluation function which helps improve the efficiency in exploring large hypothesis spaces. In this approach, the neural network does not work as a standalone ILP system, instead it is used to approximate clause evaluation scores to decide the direction of the hypothesis search. In [34], propositionalisation is used for learning first-order logic in Bayesian networks. Inspired by this work, in [14], the CILP++ system is proposed by integrating bottom clauses and rule-based approach CILP [16], referred to in Section 3.1.1.

The main advantage of propositionalisation is that it is efficient and it fits neural networks well. Also, it does not require first-order formulas to be provided as bottom clauses. However, propositionalisation has serious disadvantages. First, with func-

tion symbols, there are infinitely many ground terms. Second, propositionalization seems to generate lots of irrelevant clauses.

3.2.2 Tensorisation

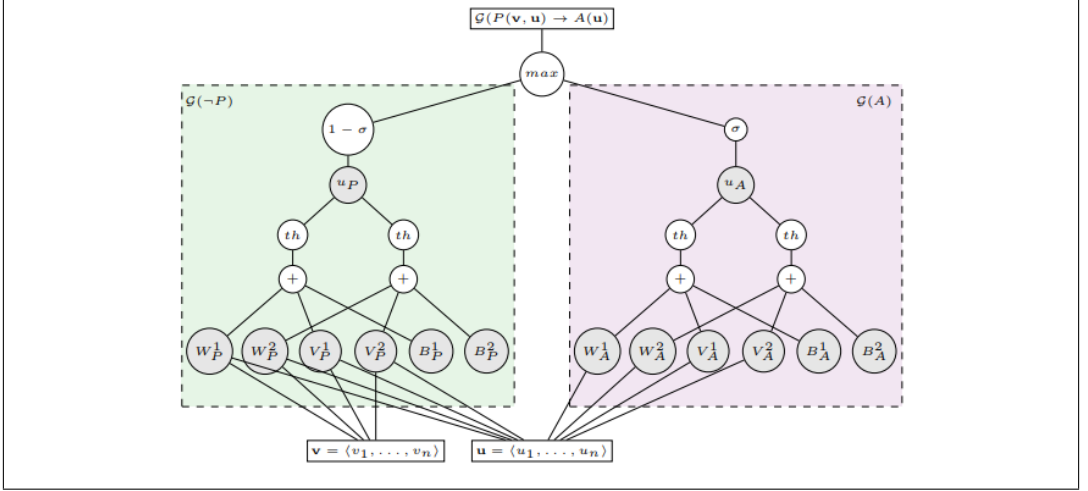


Figure 4: Logic tensor network for $P(x, y) \rightarrow A(y)$ with $\mathcal{G}(x) = \mathbf{v}$ and $\mathcal{G}(y) = \mathbf{u}$; \mathcal{G} are grounding (vector representation) for symbols in first-order language; and the tensor order in this example is 2 [40].

Tensorisation is a class of approaches that embeds first-order logic symbols such as constants, facts and rules into real-valued tensors. Normally, constants are represented as one-hot vectors (first order tensor). Predicates and functions are matrices (second-order tensor) or higher-order tensors.

In early work, embedding techniques were proposed to transform symbolic representations into vector spaces where reasoning can be done through matrix computation [3, 45, 46, 40, 39, 5, 13, 53, 12, 37]. Training embedding systems can be carried out as distance learning using backpropagation. Most research in this direction focuses on representing relational predicates in a neural network. This is known as "relational embedding" [3, 39, 45, 46]. For representation of more complex logical structures, i.e. first order-logic formulas, a system named *Logic Tensor Network* (LTN) [40] is proposed by extending *Neural Tensor Networks* (NTN) [45], a state-of-the-art relational embedding method. Figure 4 shows an example of LTN for $P(x, y) \rightarrow A(y)$. Related ideas are discussed formally in the context of constraint-based learning and reasoning [18]. Recent research in first-order logic programs has successfully exploited the advantages of distributed representations of logic sym-

bols for efficient reasoning [5], inductive programming [13, 53, 12], and differentiable theorem proving [37].

3.3 Temporal Logic

One of the earliest works on temporal logic and neural networks is CTLK, where ensembles of recurrent neural networks are set up to represent the possible world semantics of linear temporal logics [7]. With single hidden layers and semi-linear neurons, the networks can compute a fixed-point semantics of temporal logic rules. Another work on representation of temporal knowledge is proposed in *Sequential Connectionist Temporal Logic* (SCTL) [4] where CILP is extended to work with the nonlinear auto-regressive exogenous NARX network model. *Neural-Symbolic Cognitive Agents* (NSCA) represent temporal knowledge in recurrent temporal RBMs [32]. Here, the temporal logic rules are modelled in the form of recursive conjunctions represented by recurrent structures of RBMs. Temporal relational knowledge embedding has been studied recently in *Tensor Product Recurrent Neural Network* (TPRN) with applications to question-answering [30].

4 Neural-Symbolic Learning

4.1 Inductive Logic Programming

Inductive logic programming (ILP) can take advantage of the learning capability of neural-symbolic computing to automatically construct a logic program from examples. Normally, approaches in ILP are categorised into *bottom-up* and *top-down* which inspire the development of neural-symbolic approaches accordingly for learning logical rules.

Bottom-up approaches construct logic programs by extracting specific clauses from examples. After that, generalisation procedures are usually applied to search for more general clauses. This is well suited to the idea of propositionalisation discussed earlier in Section 3.2.1. For example, CILP++ [14] employed a bottom clause propositionalisation technique to construct CILP++. In [49], a system called CRILP is proposed by integrating bottom clauses generated from [14] with RBMs. However, both CILP++ and CRILP learn and fine-tune formulas at a propositional level where propositionalisation would generate a large number of long clauses resulting in very large networks. This leaves an open research question of generalising bottom clauses within neural networks that scale well and can extrapolate.

Top-down approaches, on the other hand, construct logic programs from the most general clauses and extend them to be more specific. In neural-symbolic terms, the

most popular idea is to take advantage of neural networks’ learning and inference capabilities to fine-tune and test the quality of rules. This can be done by replacing logical operations by differentiable operations. For example, in Neural Logic Programming (NLP) [53], learning of rules are based on the differentiable inference of TensorLog [5]. Here, matrix computations are used to soften logic operators where the confidence of conjunctions and confidence of disjunctions are computed as product and sum, respectively. NLP generate rules from facts, starting with the most general ones. In Differentiable Inductive Logic Programming (∂ ILP) [13], rules are generated from templates, which are assigned to parameters (weights) to make the loss function between actual conclusions and predicted conclusions from forward chaining differentiable. In [37], Neural Theorem Prover (NTP) is proposed by extending the backward chaining method to be differentiable. It shows that latent predicates from rule templates can be learned through optimisation of their distributed representations. Different from [53, 13, 37] where clauses are generated and then softened by neural networks, in Neural Logic Machines (NLM) [12] the relation of predicates is learned by a neural network where input tensors represent facts (predicates of different arities) from a knowledge base and output tensors represent new facts.

4.2 Horizontal Hybrid Learning

Effective techniques such as deep learning usually require large amounts of data to exhibit statistical regularities. However, in many cases where collecting data is difficult a small dataset would make complex models more prone to overfitting. When prior knowledge is provided, e.g. from domain experts, a neural-symbolic system can offer the advantage of generality by combining logical rules/formulas with data during learning, while at the same time using the data to fine-tune the knowledge. It has been shown that encoding knowledge into a neural network can result in performance improvements [6, 11, 47, 49]. Also, it is evident that using symbolic knowledge can help improve the efficiency of neural network learning [6, 14]. Such effectiveness and efficiency are obtained by encoding logical knowledge as controlled parameters during the training of a model. This technique, in general terms, has been known as learning with logical constraints [18]. Besides, in the case of lacking prior knowledge one can apply the idea of neural-symbolic integration for knowledge transfer learning [48]. The idea is to extract symbolic knowledge from a related domain and transfer it to improve the learning in another domain, starting from a network that does not necessarily have to be instilled with background knowledge. Self-transfer with symbolic-knowledge distillation [22] is also useful as it can enhance several types of deep networks such as convolutional neural networks and recurrent

neural networks. Here, symbolic knowledge is extracted from a trained network called “teacher” which then would be used to encoded as regularizers to train a “student” network in the same domain.

4.3 Vertical Hybrid Learning

Studies in neuroscience show that some areas in the brain are used for processing input signals e.g. visual cortices for images [19, 35], while other areas are responsible for logical thinking and reasoning [41]. Deep neural networks can learn high level abstractions from complex input data such as images, audio, and text, which should be useful at making decisions. However, despite that optimisation process during learning being mathematically justified, it is difficult for humans to comprehend how a decision has been made during inference time. Therefore, placing a logic network on top of a deep neural network to learn the relations of those abstractions, can help the system to be able to explain itself. In [11], a Fast-RCNN [17] is used for bounding-box detection of parts of objects and on top of that, a Logic Tensor Network is used to reason about relations between parts of objects and types of such objects. In such work, the perception part (Fast-RCNN) is fixed and learning is carried out in the reasoning part (LTN). In a related approach, called DeepProbLog, end-to-end learning and reasoning have been studied [26] where outputs of neural networks are used as "neural predicates" for ProbLog [9].

5 Neural-symbolic Reasoning

Reasoning is an important feature of a neural-symbolic system and has recently attracted much attention from the research community [13]. Various attempts have been made to perform reasoning within neural networks, both model-based and theorem proving approaches. In neural-symbolic integration the main focus is the integration of reasoning and learning, so that a model-based approach is preferred. Most theorem proving systems based on neural networks, including first-order logic reasoning systems such as SHRUTI [52], have been unable to perform learning as effectively as end-to-end differentiable learning systems. On the other hand, model-based approaches have been shown implementable in neural networks in the case of nonmonotonic, intuitionistic and propositional modal logic, as well as abductive reasoning and other forms of human reasoning [1, 4]. As a result, the focus of neural-symbolic computation has changed from performing symbolic reasoning in neural networks, such as for example implementing the logical unification algorithm in a neural network, to the combination of learning and reasoning, in some cases with a much more loosely-defined approach rather than full integration, whereby a

hybrid system will contain different components which may be neural or symbolic and which communicate with each other.

5.1 Forward and Backward chaining

Forward chaining and backward chaining are two popular inference techniques for logic programs and other logical systems. In the case of neural-symbolic systems forward and backward chainings are both in general implemented by feedforward inference.

Forward chaining generates new facts from the head literals of the rules using existing facts in the knowledge base. For example, in [32], a “Neural-symbolic Cognitive Agent” shows that it is possible to perform online learning and reasoning in real-world scenarios, where temporal knowledge can be extracted to reason about driving skills [32]. This can be seen as forward chaining over time. In ∂ ILP [13], a differentiable function is defined for each clause to carry out a single step of forward chaining. Similar to this, NLM [12] employs neural networks as a differentiable chain for forward inference. Different from ∂ ILP, NLM represent the outputs and inputs of neural networks as grounding tensors of predicates for existing facts and new facts respectively.

Backward chaining, on the other hand, searches backward from a goal in the knowledge base to determine whether a query is derivable or not. This form a tree search starts from the query and expands further to the literals in the body of the rules whose heads match the query. TensorLog [5] implements backward chaining using neural networks as symbols. The idea is based on stochastic logic programs [29], and soft logic is applied to transform the hypothesis search into a chain of matrix operations. In NTP, a neural system is constructed recursively for backward chaining and unification where AND and OR operators are represented as networks. In general, backward (goal-directed) reasoning is considerably harder to achieve in neural networks than forward reasoning. This is another current line of research within neuro-symbolic computation and AI.

5.2 Approximate Satisfiability

Inference in the case of logic programs with arbitrary formulas is more complex. In general, one may want to search over the hypothesis space to find a solution that satisfies (mostly) the formulas and facts in the knowledge base. Exact inference, that is, reasoning maximising satisfiability, is NP-hard. For this reason, some neural-symbolic systems offer a mechanism of approximate satisfiability. Tensor logic networks are trained to approximate the best satisfiability [40] making inference effi-

cient with feedforward propagation. This has made LTNs applicable successfully to the Pascal data set and image understanding [11]. Penalty logic shows an equivalence between minimising violation and minimising energy functions of symmetric connectionist networks [33]. Confidence rules, another approximation approach, shows the relation between sampling in restricted Boltzmann machines and search for truth-assignments which maximise satisfiability. The use of confidence rules also allows one to measure how confident a neural network is in its own answers. Based on that, neural-symbolic system “confidence rule inductive logic programming (CRILP)” was constructed and applied to inductive logic programming [49].

5.3 Relationship reasoning

Relational embedding systems have been used for reasoning about relationships between entities. Technically, this has been done by searching for the answer to a query that gives the highest grounding score [3, 2, 45, 46]. Deep neural networks are also employed for visual reasoning where they learn and infer relationships and features of multiple objects in images [39, 54, 27].

6 Neural-symbolic Explainability

The (re)emergence of deep networks has again raised the question of explainability. The complex structure of a deep neural network turns them into a powerful learning system if one can correctly engineer its components such as type of hidden units, regularisation and optimization methods. However, limitations of some AI applications have heightened the need for explainability and interpretability of deep neural networks. More importantly, besides improving deep neural networks for better applications one should also look for the benefits that deep networks can offer in terms of knowledge acquisition.

recognition [42].

6.1 Natural Language Generation

For explainability purposes, another approach couples a deep network with sequence models to extract natural language knowledge [21]. In [3], instead of investigating the parameters of a trained model, relational knowledge extraction is proposed where predicates are obtained by performing inference of a trained embedding network on text data.

6.2 Program Synthesis

In the field of Program Induction, neuro-symbolic program synthesis (NSPS) has been proposed to construct computer programs on an incremental fashion using a large amount of input-output samples [31]. A neural network is employed to represent partial trees in a domain-specific language where tree nodes, symbols and rules are vector representations. Explainability can be achieved through the tree-based structure of the network. Again, this shows that the integration of neural networks and symbolic representation is indeed a solution for both scalability and explainability.

7 Conclusions

In this paper, we highlighted the key ideas and principles of neural-symbolic computing. In order to do so, we illustrated the main methodological approaches which allow for the integration of effective neural learning with sound symbolic-based, knowledge representation and reasoning methods. One of the principles we highlighted in the paper is the sound mapping between symbolic rules and neural networks provided by neural-symbolic computing methods. This mapping allows several knowledge representation formalisms to be used as background knowledge for potentially large-scale learning and efficient reasoning. This interplay between efficient neural learning and symbolic reasoning opens relevant possibilities towards richer intelligent systems. The comprehensibility and compositionality of neural-symbolic systems, offered by building networks with a logical structure, allows for integrated learning and reasoning under different logical systems. This opens several interesting research lines, in which learning is endowed with the sound semantics of diverse logics. This, in turn, contributes towards the development of explainable and accountable AI and machine learning-based systems and tools.

[illegible]

References

- [1] S. Bader and P. Hitzler. Dimensions of neural-symbolic integration: a structured survey. In S. Artemov, H. Barringer, A. d’Avila Garcez, L. Lamb, and J. Woods, editors, *We Will Show Them! Essays in Honour of Dov Gabbay*, 2005.
- [2] A. Bordes, X. Glorot, J. Weston, and Y. Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *AISTATS*, pages 127–135, 2012.
- [3] A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning structured embeddings of knowledge bases. In *AAAI’11*, 2011.
- [4] R. Borges, A. d’Avila Garcez, and L.C. Lamb. Learning and representing temporal knowledge in recurrent networks. *IEEE Trans. Neural Networks*, 22(12):2409–2421, 2011.
- [5] William W. Cohen, Fan Yang, and Kathryn Mazaitis. Tensorlog: Deep learning meets probabilistic dbs. *CoRR*, abs/1707.05390, 2017.
- [6] A. d’Avila Garcez and G. Zaverucha. The connectionist inductive learning and logic programming system. *Appl. Intelligence*, 11(1):59–77, 1999.
- [7] A.S. d’Avila Garcez and L.C. Lamb. Reasoning about time and knowledge in neural symbolic learning systems. In *NIPS*, pages 921–928, 2003.
- [8] A.S. d’Avila Garcez and L.C. Lamb. A connectionist computational model for epistemic and temporal reasoning. *Neur. Computation*, 18(7):1711–1738, 2006.
- [9] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. Problog: A probabilistic prolog and its application in link discovery. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI’07*, pages 2468–2473, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.
- [10] Frank DiMaio and Jude Shavlik. Learning an approximation to inductive logic programming clause evaluation. In Rui Camacho, Ross King, and Ashwin Srinivasan, editors, *Inductive Logic Programming*, pages 80–97, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [11] I. Donadello, L. Serafini, and A. S. d’Avila Garcez. Logic tensor networks for semantic image interpretation. In *IJCAI-17*, pages 1596–1602, 2017.
- [12] Honghua Dong, Jiayuan Mao, Tian Lin, Chong Wang, Lihong Li, and Denny Zhou. Neural logic machines. 2019.
- [13] R. Evans and E. Grefenstette. Learning explanatory rules from noisy data. *JAIR*, 61:1–64, 2018.
- [14] M. França, G. Zaverucha, and A. Garcez. Fast relational learning using bottom clause propositionalization with artificial neural networks. *Mach. Learning*, 94(1):81–104, 2014.
- [15] A. d’Avila Garcez, L.C. Lamb, and D.M. Gabbay. *Neural-Symbolic Cognitive Reasoning*. Springer, 2009.
- [16] A.S. d’Avila Garcez, D. Gabbay, and K.. Broda. *Neural-Symbolic Learning System: Foundations and Applications*. Springer, 2002.

- [17] Ross Girshick. Fast r-cnn. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1440–1448, Washington, DC, USA, 2015. IEEE Computer Society.
- [18] Marco Gori. *Machine Learning: A Constraint-Based Approach*. Morgan Kaufmann, 2018.
- [19] Kalanit Grill-Spector and Rafael Malach. The human visual cortex. *Annual Review of Neuroscience*, 27(1):649–677, 2004.
- [20] B. Hammer and P. Hitzler, editors. *Perspectives of Neural-Symbolic Integration*. Springer, 2007.
- [21] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *Computer Vision - ECCV 2016 - 14th European Conference*, pages 3–19, 2016.
- [22] Z. Hu, X. Ma, Z. Liu, E. Hovy, and E. Xing. Harnessing deep neural networks with logic rules. In *ACL*, 2016.
- [23] R. Khardon and D. Roth. Learning to reason. *J. ACM*, 44(5), 1997.
- [24] L.C. Lamb, R.V. Borges, and A.S. d’Avila Garcez. A connectionist cognitive model for temporal synchronisation and learning. In *AAAI*, pages 827–832, 2007.
- [25] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [26] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 3749–3759. Curran Associates, Inc., 2018.
- [27] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In *International Conference on Learning Representations*, 2019.
- [28] Stephen Muggleton. Inverse entailment and prolog. *New Generation Computing*, 13(3):245–286, Dec 1995.
- [29] Stephen Muggleton. Stochastic logic programs. In *New Generation Computing*. Academic Press, 1996.
- [30] H. Palangi, P. Smolensky, X. He, and L. Deng. Question-answering with grammatically-interpretable representations. In *AAAI*, 2018.
- [31] E. Parisotto, A.-R. Mohamed, R. Singh, L. Li, D. Zhou, and P. Kohli. Neuro-symbolic program synthesis. In *ICLR*, 2017.
- [32] L. de Penning, A. d’Avila Garcez, L.C. Lamb, and J-J. Meyer. A neural-symbolic cognitive agent for online learning and reasoning. In *IJCAI*, pages 1653–1658, 2011.
- [33] G. Pinkas. Reasoning, nonmonotonicity and learning in connectionist networks that capture propositional knowledge. *Artif. Intell.*, 77(2):203–247, 1995.
- [34] Cristiano Grijó Pitangui and Gerson Zaverucha. Learning theories using estimation distribution algorithms and (reduced) bottom clauses. In Stephen H. Muggleton, Alireza Tamaddoni-Nezhad, and Francesca A. Lisi, editors, *Inductive Logic Programming*, pages

- 286–301, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [35] Tomaso A. Poggio and Fabio Anselmi. *Visual Cortex and Deep Networks: Learning Invariant Representations*. The MIT Press, 1st edition, 2016.
 - [36] H. Poon and P. Domingos. Sum-product networks: A new deep architecture. In *2011 IEEE Int. Conf. on Computer Vision Workshops (ICCV Workshops)*, 2011.
 - [37] Tim Rocktäschel and Sebastian Riedel. Learning knowledge base inference with neural theorem provers. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 45–50, San Diego, CA, June 2016. Association for Computational Linguistics.
 - [38] S.J. Russell, S. Hauert, R. Altman, and M. Veloso. Ethics of artificial intelligence: Four leading researchers share their concerns and solutions for reducing societal risks from intelligent machines. *Nature*, 521:415–418, 2015.
 - [39] A. Santoro, D. Raposo, D. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017.
 - [40] Luciano Serafini and Artur S. d’Avila Garcez. Learning and reasoning with logic tensor networks. In *AI*IA*, pages 334–348, 2016.
 - [41] Ehsan Shokri-Kojori, Michael A. Motes, Bart Rypma, and Daniel C. Krawczyk. The network architecture of cortical processing in visuo-spatial reasoning. *Scientific Reports*, 2, 2012.
 - [42] Z. Si and S. C. Zhu. Learning and-or templates for object recognition and detection. *IEEE Trans. Pattern Analysis and Mach. Intell.*, 35(9):2189–2205, Sept 2013.
 - [43] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(354), 2017.
 - [44] P. Smolensky. Constituent structure and explanation in an integrated connectionist/symbolic cognitive architecture. In *Connectionism: Debates on Psychological Explanation*. 1995.
 - [45] R. Socher, D. Chen, C. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion. In *NIPS*, pages 926–934. 2013.
 - [46] I. Sutskever and G. Hinton. Using matrices to model symbolic relationship. In *NIPS*. 2009.
 - [47] G. Towell and J. Shavlik. Knowledge-based artificial neural networks. *Artif. Intel.*, 70:119–165, 1994.
 - [48] S. Tran and A. Garcez. Deep logic networks: Inserting and extracting knowledge from deep belief networks. *IEEE T. Neur. Net. Learning Syst.*, (29):246–258, 2018.
 - [49] Son N. Tran. Propositional knowledge representation and reasoning in restricted boltzmann machines. *CoRR*, abs/1705.10899, 2018.
 - [50] L. Valiant. Knowledge infusion. In *AAAI*, 2006.
 - [51] L.G. Valiant. Three problems in computer science. *J. ACM*, 50(1):96–99, 2003.
 - [52] Carter Wendelken and Lokendra Shastri. Multiple instantiation and rule mediation in

- SHRUTI. *Connect. Sci.*, 16(3):211–217, 2004.
- [53] Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2319–2328. Curran Associates, Inc., 2017.
- [54] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1031–1042. Curran Associates, Inc., 2018.