# Perspectives of Neuro–Symbolic Integration – Extended Abstract –

Kai-Uwe Kühnberger, Helmar Gust, Peter Geibel

Institute of Cognitive Science, University of Osnabrück
49076, Osnabrück, Albrechtstr. 28, Germany
{kkuehnbe,hgust,pgeibel}@uos.de

**Abstract.** There is an obvious tension between symbolic and subsymbolic theories, because both show complementary strengths and weaknesses in corresponding applications and underlying methodologies. The resulting gap in the foundations and the applicability of these approaches is theoretically unsatisfactory and practically undesirable. We sketch a theory that bridges this gap between symbolic and subsymbolic approaches by the introduction of a Topos-based semi-symbolic level used for coding logical first-order expressions in a homogeneous framework. This semi-symbolic level can be used for neural learning of logical first-order theories. Besides a presentation of the general idea of the framework, we sketch some challenges and important open problems for future research with respect to the presented approach and the field of neuro-symbolic integration, in general.

**Keywords.** Neuro–Symbolic Integration, Topos Theory, First–Order Logic

## 1 Introduction

It is generally accepted that there is a tension between symbolic and subsymbolic approaches for modeling cognitive abilities. Whereas both approaches have complementary strengths and weaknesses in application domains, these differences can also be recognized in the underlying methodological foundations. For example, whereas symbolic theories are usually based on logical or algebraic frameworks, sub-symbolic models are often mathematically based on analytic methods. With respect to possible application domains, it turns out that areas in which symbolic theories are strongly used, like reasoning, problem solving, planning etc., usually sub-symbolic modelings do have their problems. On the other hand, learning, motor control, vision etc. are strong domains for sub-symbolic approaches, for which symbolic theories show some deficits. Table 1 summarizes some important differences between these two approaches. Obviously, the table only mentions some important differences between the underlying theories. Furthermore, not all distinction are undisputable. For example, whether all subsymbolic approaches can count as biologically inspired is probably not generally accepted and whether it is possible to make a precise distinction between application domains of symbolic and subsymbolic theories is also unclear. Nevertheless,

Dagstuhl Seminar Proceedings 08041
Recurrent Neural Networks- Models, Capacities, and Applications
http://drops.dagstuhl.de/opus/volltexte/2008/1422

**Table 1.** Important differences between symbolic and subsymbolic approaches

| | **Symbolic Approaches** | **Subsymbolic Approaches** |
|---|---|---|
| **Methods** | (Mostly) logical and/or algebraic | (Mostly) analytic |
| **Strengths** | Productivity, Recursion Principle, Compositionality | Robustness, Learning Ability, Parsimony, Adaptivity |
| **Weaknesses** | Consistency Constraints, Lower Cognitive Abilities | Opaqueness, Higher Cognitive Abilities |
| **Applications** | Reasoning, Problem Solving, Planning etc. | Learning, Motor Control, Vision etc. |
| **Relation to CogSci** | Not Biologically Inspired | Biologically Inspired |
| **Other Features** | Crisp | Fuzzy, Continuous |

the table shows a tendency of how to classify the two types of modeling in an overall picture.

In a series of papers (cf. [1], [2], [3]), the authors propose a theory for neuro-symbolic integration. This theory is crucially based on the idea of translating logical input into a variable-free representation using Topos theory. This semi-symbolic level can be used for generating a homogeneous input for a neural network.[1]

## 2 The General Idea of Using Topos Theory for Neuro-Symbolic Integration

In this section, we sketch the overall idea of the approach. Figure 1 summarizes the architecture of the system. The following list describes informally the main modules of the systems:

- As input a first–order logical theory $T$ of a language $\mathcal{L}$ is given.
- $T$ is translated into a variable-free representation in a topos, i.e. a category in the sense of category theory with "nice" properties (cf. [4]). The result is a representation in terms of commuting diagrams, i.e. objects and arrows instead of the original symbolic interpretation. The logical theory itself is implicitly coded by possibilities of constructing new diagrams in the topos (limit constructions, sub-object classifier, exponents etc.).
- An algorithm is generating equations of the normal form $f \circ g = h$ and inequations of the form $f \neq g$. These equations and inequations correspond to equations and inequations of arrows in the topos. Due to the fact that a topos allows universal constructions, these equations and inequations can be automatically generated.
- Objects and arrows are considered to be atomic in category theory. Therefore, there are no crucial restrictions concerning the choice of the representations. This means it is possible to choose vectors of the vector space

[1] The notion "semi-symbolic level" is used for denoting the Topos representation of logical input due to the fact that neither all symbols have a Topos counterpart (e.g. quantifiers are not represented as Topos entities) nor does each Topos entity correspond to a symbol or a symbolic expression.

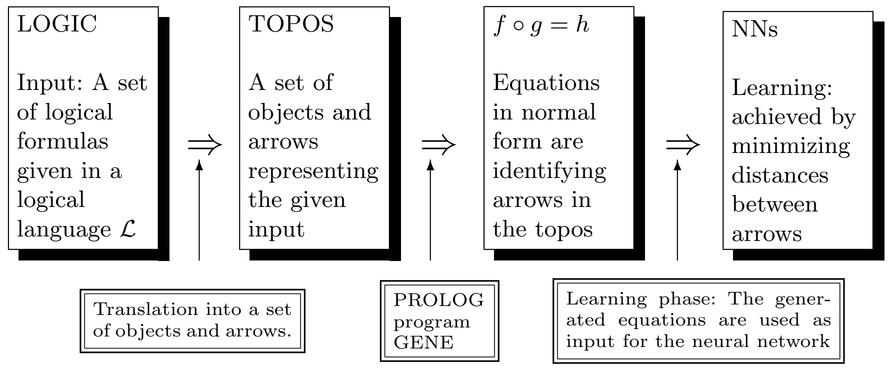


**Fig. 1.** General architecture: The input, given as a set of logical axioms, is translated to Topos entities (objects and arrows). Topos constraints allow the generation of equations in normal form – practically realized by a PROLOG program GENE – constraining the equality or inequality of arrows. These are represented by real-valued vectors and function as input for neural learning. Backpropagation is used to learn the representations of these entities, such that the truth conditions of the axioms are satisfied.

$\mathbb{R}^n$ as representations of objects and arrows. The resulting equations and inequations can be used in order to train a neural network.

- Each arrow is represented by three vectors: (1) the domain of the arrow, (2) the representation of the arrow itself (i.e. its identifier), and (3) the codomain of the arrow.
- The composition of two arrows (resulting in a further arrow) is represented by five vectors, due to the fact that the codomain of the first arrow equals the domain of the second arrow. The composition of two arrows is used as input for the network.
- The output of the network is the representation of the resulting arrow based on the composition of the two input arrows. This result is compared with the right side of the equation, such that backpropagation of the error can be performed resulting in an adaptation of the weights of the network. It should be noticed that besides learning the composition operation, it is also important to learn the representations and the domains and codomains of arrows themselves.

– The result of the approach is the learning of a model of a logical theory $T$, namely the representation of the symbols and expressions of $\mathcal{L}$, such that the truth conditions of the axioms of $T$ are satisfied.

It should be emphasized that the sketched approach does not only allow to learn the representation of the logical input, but also forecasts the truth values of any possible formula of $\mathcal{L}$ (based on the given set of axioms $T$). In other words, the approach approximates a semantic entailment concept.

In example evaluations, the framework was applied to different problems taken from the domain of theorem proving and showed promising (although clearly not optimal) results [2]. As a matter of fact, relatively simple theories can be learned rather efficiently and convincingly. Nevertheless, it is currently not possible to learn benchmark problems of theorem proving in a way that neural learning can be considered as an alternative to symbolic theorem provers. A convincing application of neuro-symbolic integration that is competitive to symbolic approaches remains an open problem for the future.

## 3 Challenges for Neuro–Symbolic Integration

### 3.1 Specific Challenges for the Presented Approach

The sketched ideas provide a first step towards a theory of neuro-symbolic integration. There is a bunch of open problems we would like to discuss for future research. We start with some specific open problems related to the approach presented here:

– Although certain properties of the learned models can be identified [3], an abstract characterization of the learned models is missing. In particular, it would be desirable to specify theoretically the class of models that can be learned with the sketched Topos-based approach.
– A technical question concerns the relation between the complexity of the logical input theory $T$ and the number of equations that are needed in order to approximate a model to a certain degree. An example for a measure for the "degree" to which a model of $T$ is learned could be specified by the number of times a certain deduction operator is applied.
– A theoretical result that characterizes under which constraints the convergence of the learning procedure is guaranteed is still missing.

### 3.2 Application Domains

The problems summarized in the list of Subsection 3.1 above concern mainly theoretical problems of the presented approach. There are also general challenges that concern every model of neuro-symbolic integration. One is the question of appropriate application domains.

– More applications are needed, not only from the theorem proving domain, but also from other domains.

This problem generalizes to every theory of neuro-symbolic integration, due to the fact that neuro-symbolic integration is currently in a state, where either the application problems are essentially propositional in nature[2], or the applica-

[2] A prominent example collecting a variation of attempts to model extensions of propositional logic in terms of modal, tense, non-monotonic etc. logic is presented in [5].

bility to predicate logic problems is limited[3]. As a consequence neuro-symbolic integration needs to prove that not only theoretically it can be shown that neural learning of logical theories is possible, but also that there are practical applications for such an approach. Such potential applications can be used for an independent justification and a proof of concept of neuro-symbolic integration, in general.

We think that – besides theorem proving – there are several promising domains that can be considered as a test scenario for neuro-symbolic integration. One idea of such a test domain was already mentioned in [7], namely ontology learning. Although there is a variety of logics that were proposed for representing conceptual knowledge in ontologies ranging from more or less weak description logics (e.g. OWL-based modelings) to full first-order logic (e.g. CYC)[4], the logics underlying most practical applications are rather weak. This means that one has first, a rather controlled environment and second, the information that needs to be learned is often (at least in the description logic case) represented in a two-variable logic fragment of predicate logic. In other words, the prefixes of quantifiers in possible formulas that function as axioms of theories and need to be considered and learned are limited to length 1. At least for the present Topos-based approach this reduces the complexity of the learning process significantly, because of an exponential growth of the number of equations that need to be learned for approximating a model, if quantifier prefixes increase in the input formulas.

A second scenario for a practical application of neuro-symbolic integration is the planning domain. In real-world applications, it is a matter of fact that agents need to decide in real-time which actions they should perform next. Often, the reasoning device has certain problems in performing such decisions appropriately. A trained neural network in the sense described in this extended abstract should, in principle, be able to perform such decisions under all external time constraints due to the fact that there is not much processing necessary – at least in comparison to symbolic reasoning devices.

If it were possible to show that a particular approach for neural-symbolic integration can prove to be successfully and robustly applied to ontology learning or in the planning domain (or any other interesting domain that is not mentioned here), such that the usage of a neuro-symbolic integration device would have advantages in comparison to other more traditional techniques, then not only a proof of concept, but a realistic application scenario were available.

### 3.3 Learning

Learning a model of a logical theory as described here or learning a deduction operator of a logic program as described in [8] is in a certain sense a non-standard

[3] According to our knowledge there is only the approach presented in [6] that tries to learn full first-order theories with neural means. The presented approach as well as the theory described in [6] is currently not able to compete against symbolic reasoning algorithms

[4] Cf. http://www.cyc.com/.

usage of learning mechanisms. Usually learning is concerned with finding generalizations (in form of hypotheses or patterns) of given input data that allow to predict how unseen examples can be classified, interpreted etc. A challenge for neuro-symbolic integration concerns the development of finding algorithmic solutions on the neural basis to cover a whole range of learning strategies, in particular, to allow also the learning of generalizing hypotheses in a logical setting. The prototypical example of a symbolic learning mechanism in this field is inductive logic programming (ILP).

Despite some exceptions (an example is [9]) not much attention has been paid so far to inductive reasoning mechanisms in the neuro-symbolic integration community. Nevertheless it would be a natural idea to extend neural learning devices originally designed for learning logical deductions to inductive reasoning as well. A system that convincingly performs deductions and inductions in a uniform framework can be considered as an integration architecture for several forms of reasoning. In order to push artificial intelligence closer into the direction of a generalizing theory for modeling certain cognitive abilities, such an expansion seems to be inevitable.

## References

1. Gust, H., Kühnberger, K.U., Geibel, P.: Learning and memorizing models of logical theories in a hybrid learning device. In: Proceedings of ICONIP 2007. (2008)
2. Gust, H., Kühnberger, K.U., Geibel, P.: Learning models of predicate logical theories with neural networks based on topos theory. In Hammer, B., Hitzler, P., eds.: Perspectives on Neural-Symbolic Integration. Volume 77 of SCI. Springer (2007) 233–264
3. Gust, H., Kühnberger, K.U.: Learning symbolic inferences with neural networks. In Bara, B., Barsalou, L., Bucciarelli, M., eds.: CogSci 2005: XXVII Annual Conference of the Cognitive Science Society, Lawrence Erlbaum (2005) 875–880
4. Goldblatt, R.: Topoi : The Categorial Analysis of Logic. Volume 98 of Studies in Logic and the Foundations of Mathematics. North-Holland (1979)
5. d'Avila Garcez, A., Lamb, L., Gabbay, D.: Neural-Symbolic Cognitive Reasoning. Cognitive Technologies. Springer (2008)
6. Bader, S., Hitzler, P., Hölldobler, S., Witzel, A.: A fully connectionist model generator for covered first-order logic programs. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence, AAAI Press, Menlo Park CA (2007) 666–671
7. Hitzler, P., Bader, S., d'Avila Garcez, A.: Ontology leaning as a use case for neural-symbolic integration. In d'Avila Garcez, A., Hitzler, P., Ellman, J., eds.: Proceedings of the IJCAI-05 Workshop on Neural-Symbolic Learning and Reasoning. (2005)
8. Hitzler, P., Hölldobler, S., Seda, A.: Logic programs and connectionist networks. Journal of Applied Logic **2(3)** (2004) 245–272
9. d'Avila Garcez, A., Zaverucha, G.: The connectionist inductive learning and logic programming system. Applied Intelligence. Special Issue on Neural Networks and Structured Knowledge **11(1)** (1999) 59–77