The Integration of Connectionism and First-Order Knowledge Representation and Reasoning as a Challenge for Artificial Intelligence (extended abstract)

Sebastian Bader[*], Pascal Hitzler[†], Steffen Hölldobler[†]

International Center for Computational Logic
Technische Universität Dresden, Germany

Intelligent systems based on first-order logic on the one hand, and on artificial neural networks (also called connectionist systems) on the other, differ substantially. Logic programs, for example, are highly recursive and well understood from the perspective of knowledge representation and reasoning: Their semantics has been studied extensively in the past, which makes it easy to encode problem specifications directly as programs. One reason for the success of artificial neural networks lies in the fact that they can be trained using raw data, and in some problem domains the generalization from the raw data made during the learning process turns out to be highly adequate for the problem at hand, even if the training data contains noise. Successful architectures, however, often do not use recursive (or recurrent) structures. Furthermore, the knowledge encoded by a trained neural network is only very implicitly represented, and no fully satisfactory methods for extracting this knowledge in symbolic form are currently known.

It would be very desirable to combine the robust neural networking machinery with symbolic knowledge representation and reasoning paradigms like logic programming in such a way that the strengths of either paradigm will be retained. Current state-of-the-art research, however – for surveys see [2, 3, 4] – fails by far to achieve this ultimate goal: We are unaware of any connectionist system which can handle structured objects and structure-sensitive processes in a satisfying way. First-order logic systems were designed to cope with such objects and processes and, consequently, it is a long-standing research goal to combine the advantages of connectionist and first-order logic systems in a single system.

Earlier attempts to integrate logic and connectionist systems have mainly been restricted to propositional logic, or to first-order logic without function symbols. Without the restriction to the finite case (including propositional logic and first-order logic without function symbols), the task becomes much harder due to the fact that the underlying language is infinite but shall be encoded using networks with a finite number of nodes. One of the few approaches for overcoming this problem is based on a proposal by Hölldobler et al. [7], spelled out first for the propositional case in [6], and reported also in [5]. It is based on the idea that logic programs can be represented by their associated single-step or immediate consequence operators. Such an operator can then

[*]This author acknowledges support by the DFG within the postgraduate programme 334 "Specification of discrete processes and systems of processes by operational models and logics".

[†]These authors acknowledge support by the Boole Centre for Research in Informatics at the National University of Ireland, University College Cork, for presenting this paper.

be mapped to a function on the real numbers, which can under certain conditions in turn be encoded or approximated e.g. by feedforward networks with sigmoidal activation functions. Turning such feedforward networks into recurrent ones allows for the computation (in the propositional case) or approximation (in the first-order case) of the semantics of the logic program.

The purpose of our presentation is twofold. First, we will give an overview of recent progress made by us in the representation of first-order logic programs by connectionist systems [5, 1]. We will then discuss in detail the following questions which we find of central importance in order to advance towards an integration of logic and connectionism.

1. How can first-order terms be represented and manipulated in a connectionist system?
2. How can first-order rules be extracted from a connectionist network?
3. How can distributed knowledge representation in connectionist networks be understood from a symbolic perspective?
4. How can established learning algorithms like backpropagation be combined with symbolic knowledge representation in connectionist systems?
5. How can multiple instances of first-order rules be represented in a connectionist system?
6. How can insights from neuroscience be used to design integrated systems which are biologically feasible?
7. What is the exact relationship between neural-symbolic integration and chaos theory? Can this be exploited?
8. What does a theory for the integration of logic and connectionist systems look like?
9. Can such a theory be applied in a real domain outperforming conventional approaches?

References

[1] Sebastian Bader and Pascal Hitzler. Logic programs, iterated function systems, and recurrent radial basis function networks. *Journal of Applied Logic*, 2004. To appear.

[2] Anthony Browne and Ron Sun. Connectionist inference models. *Neural Networks*, 14(10):1331–1355, 2001.

[3] Artur S. d'Avila Garcez, Krysia B. Broda, and Dov M. Gabbay. *Neural-Symbolic Learning Systems — Foundations and Applications*. Perspectives in Neural Computing. Springer, Berlin, 2002.

[4] Hans W. Güsgen and Steffen Hölldobler. Connectionist inference systems. In Bertram Fronhöfer and Graham Wrightson, editors, *Parallelization in Inference Systems*, volume 590 of *Lecture Notes in Artificial Intelligence*, pages 82–120. Springer, Berlin, 1992.

[5] Pascal Hitzler, Steffen Hölldobler, and Anthony K. Seda. Logic programs and connectionist networks. *Journal of Applied Logic*, 2004. to appear.

[6] Steffen Hölldobler and Yvonne Kalinke. Towards a massively parallel computational model for logic programming. In *Proceedings of the ECAI94 Workshop on Combining Symbolic and Connectionist Processing*, pages 68–77. ECCAI, 1994.

[7] Steffen Hölldobler, Yvonne Kalinke, and Hans-Peter Störr. Approximating the semantics of logic programs by recurrent neural networks. *Applied Intelligence*, 11:45–58, 1999.