Abductive reasoning in neural-symbolic systems

Andrzej Gajda, Dominika Gorol
Department of Logic and Cognitive Science
Institute of Psychology
Adam Mickiewicz University in Poznań

Peirce [8, 5.189] proposed the following scheme to describe abducive reasoning: from the observed phenomenon A and from the known rule *if H then A*, infer H. This scheme can be formalised in may different ways. One of the most popular approaches amongst computer scientists is the Abductive Logic Programming framework [6], where the knowledge base is represented by a logic program (i.e. a set of clauses of the form $a_0 \leftarrow a_1, \ldots, a_n$, where $a_0, \ldots, a_n$ are atoms) and the set of integrity constraints (i.e. conditions that cannot be violated), and the phenomenon A that needs to be *explained* is a set of atomic formulas. The abductive procedure starts with the generation of one additional set of atoms that are called abducibles, i.e. atoms that occur in the logic program and potentially can form the explanation for the phenomenon A. During the abductive procedure the set of abducibles is narrowed down to the set of atoms that are necessary to explain the phenomenon A.

The aim of our talk is to propose a different interpretation of the Peirce's scheme of abductive reasoning. We employ the algorithmic point of view proposed by Gabbay and Woods where an abductive hypothesis (or abducible) H "is legitimately dischargeable to the extent to which it makes it possible to prove (or compute) from a database a formula not provable (or computable) from it as it is currently structured" [2, p. 28]. Our motivation is consist of three main reasons. Firstly, we want to be able to express the phenomenon A that we want to explain not only as a set of atoms, but also as a clause. The second reason also concerns the restriction imposed on the phenomenon A, namely we want to be able to generate abductive hypotheses also in the case when A contains information that is not present in the logic program that represents the knowledge base. Finally, instead of the generation of the set of abducibles at the beginning of the abductive procedure, we want to obtain an abductive hypothesis from the knowledge base and the abductive goal alone. In addition, the abductive hypothesis should fulfil certain criteria (e.g. it should be consistent with the knowledge base, the hypothesis alone should not entail the phenomenon A, and the abductive hypothesis should be *possibly small* [7]).

As a general framework the *Connectionist-Inductive Learning and Logic Programming* system [4] was chosen, where the knowledge base and the phenomenon to be explained can be represented in the formal language of logic programs, and the process of abductive hypotheses generation is based on the training of a neural network that represents the knowledge base and the abductive goal. The scheme of the abductive procedure can be seen in the Fig. 1. The logic program $\mathcal{P}$ that represents the knowledge base and the abductive goal is translated into a neural network $\mathfrak{N}$ by the translation algorithm $T_{\mathcal{P}\rightarrow\mathfrak{N}}$, i.e. the modified version of the algorithm proposed by Garcez et al. [4]. The neural network $\mathfrak{N}$ is

$$\begin{tikzcd}
\begin{array}{c}\text{Logic}\\ \text{program}\\ \mathcal{P}\end{array} \arrow[r, "T_{\mathcal{P}\to\mathfrak{N}}"] \arrow[d, dotted, leftrightarrow, "\begin{array}{c}\text{Difference}\\ \mathsf{d}(\mathcal{P},\mathcal{P}')\end{array}"'] & \begin{array}{c}\text{Neural}\\ \text{network}\\ \mathfrak{N}\end{array} \arrow[d, "\begin{array}{c}\text{Backpropagation}\\ \text{training}\end{array}"] \\
\begin{array}{c}\text{Changed}\\ \text{logic program}\\ \mathcal{P}'\end{array} & \begin{array}{c}\text{Trained}\\ \text{neural network}\\ \mathfrak{N}'\end{array} \arrow[l, "T_{\mathfrak{N}\to\mathcal{P}}"']
\end{tikzcd}$$

Figure 1: Schema of the abductive procedure.

then trained by means of the Backpropagation algorithm and the training set that is generated specifically on the ground of the definition of an abductive goal and the definition of used semantics for logic programs. In the next step the trained neural network $\mathfrak{N}'$ is translated back into a logic program $\mathcal{P}'$. The difference $\mathsf{d}(\mathcal{P}, \mathcal{P}')$ between the initial logic program $\mathcal{P}$ and the obtained one $\mathcal{P}'$ is the abductive hypothesis.

In our presentation we are going to describe in details proposed abductive procedure. In addition, we are going to discuss the use of other neural-symbolic systems (e.g. [1, 3]). We are also going to compare it with two other abductive procedures that are also based on the neural-symbolic systems, i.e. one proposed by Garcez et al. [5] and the other one proposed by Dietz Saldanha et al. [1], where both approaches use the Abductive Logic Programming framework.

References

[1] Emmanuelle-Anna Dietz Saldanha, Steffen Hölldobler, Carroline Dewi Puspa Kencana Ramli, and Luis Palacios Medinacelli. A core method for the weak completion semantics with skeptical abduction. *Journal of Artificial Intelligence Research*, 63:51–86, 2018.

[2] Dov M. Gabbay and John Woods. *The Reach of Abduction. Insight and Trial.* Elsevier, London, 2005.

[3] Wang Gang. A novel neural network structure constructed according to logical relations. *arXiv preprint arXiv:1903.02683*, 2019.

[4] Artur S. d'Avila Garcez, Krysia Broda, and Dov M. Gabbay. *Neural-*

Symbolic Learning Systems: Foundations and Applications. Springer-Verlag, London, 2002.

[5] Artur S. d'Avila Garcez, Dov M. Gabbay, Oliver Ray, and John Woods. Abductive reasoning in neural-symbolic systems. *Topoi: An International Review of Philosophy*, 26(1):37–49, 2007. doi: 10.1007/s11245-006-9005-5.

[6] Antonis Kakas, Robert Kowalski, and Francesca Toni. Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770, 1992.

[7] Maciej Komosinski, Adam Kups, Dorota Leszczyńska-Jasion, and Mariusz Urbański. Identifying efficient abductive hypotheses using multi-criteria dominance relation. *ACM Transactions on Computational Logic*, 15(4), 2014.

[8] Charles S. Peirce. *Collected Works*. Harvard University Press, Cambridge, MA., 1931–1958.