Learning Where and When to Reason in Neuro-Symbolic Inference

Cristina Cornelio[1], Jan Stuehmer[2,†], Shell Xu Hu[1] and Timothy Hospedales[1]

[1]Samsung AI, Cambridge, UK

[2]Heidelberg Institute for Theoretical Studies and Karlsruhe Institute of Technology

Abstract

The imposition of hard constraints on the output of neural networks is a highly desirable capability, as it instills confidence in AI by ensuring that neural network predictions adhere to domain expertise. This area has received significant attention recently, however, current methods typically enforce constraints in a "weak" form during training, with no guarantees at inference, and do not provide a general framework for different tasks/constraint types. We approach this open problem from a neuro-symbolic perspective. Our method enhances a conventional neural predictor with a reasoning module that can correct predictions errors and a neural attention module that learns to focus the reasoning effort on potential prediction errors while leaving other outputs unchanged. This framework provides a balance between the efficiency of unconstrained neural inference and the high cost of exhaustive reasoning during inference.

Keywords

Neuro-symbolic AI, Constraints in NNs, Reinforcement learning

In this work [1] we introduce a novel approach to manage the trade-off between the cost, expressivity, and exactness of reasoning during inference. Our *Neural Attention for Symbolic Reasoning (NASR)* architecture leverages the best of neural and symbolic worlds: rather than performing inexact, inexpressive, or intractable reasoning, we first execute an efficient neural-solver to solve a task, and then delegate a symbolic solver to correct any mistakes in the predictions. The errors are identified by our novel neural-attention module that learns *when and where to reason* in order to effectively achieve high prediction accuracy and constraint satisfaction with low computation cost.

The Method. We introduce a new neuro-symbolic integration method with a novel neural-attention module (Mask-Predictor) that works with any type of constraints/rules. More formally, we consider a set of input data points ($x \in \mathcal{X}$) representing instances to solve (e.g. the picture of a partially filled Sudoku board), and, a set of multi-dimensional output data points ($y \in \mathcal{Y}$) that correspond to complete interpretable (symbolic) solutions (e.g. the symbolic representation of a completely filled Sudoku board). The collection of N of these pairs of data points will form the *task dataset* $D = \{x^i, y^i\}_{i=1}^N$. Moreover, we require that the task (e.g. completing a partially filled Sudoku board) can be expressed (fully or partially) by a set of rules $\mathcal{R}$ in the

NeSy 2023, 17th International Workshop on Neural-Symbolic Learning and Reasoning, Certosa di Pontignano, Siena, Italy

†Work done while at Samsung AI, Cambridge, UK.

c.cornelio@samsung.com (C. Cornelio); jan.stuehmer@h-its.org.com (J. Stuehmer); shell.hu@samsung.com (S. X. Hu); t.hospedales@samsung.com (T. Hospedales)

form of domain-knowledge constraints (e.g. the rules of the Sudoku game). The goal is to learn a function $f : \mathcal{X} \to \mathcal{Y}$ associating a solution to a given input instance, *and which satisfies the rules* $\mathcal{R}$. To solve this class of problems, we propose NASR, a neuro-symbolic pipeline that works as follows: an input instance is first processed by the *Neuro-Solver* that outputs an approximate solution. The solution is then analyzed by the *Mask-Predictor* that has the role of identifying the components of the Neuro-Solver predictions that do not satisfy the set of domain-knowledge constraints/rules $\mathcal{R}$. The masking output of the Mask-Predictor is then combined with the probability distribution predicted by the Neuro-Solver. This is done by deleting the wrong elements of the predictions, leaving the corresponding components "empty" (the component is filled by an additional class, indicating a masked element). This masked probability distribution is then fed to the *Symbolic-Solver* that fills the gaps with a feasible solution (satisfying the rules $\mathcal{R}$). In brief, the role of the Symbolic-Solver is to correct the Neuro-Solver prediction errors identified by the Mask-Predictor.

Results. We tested NASR on two tasks: Visual-Sudoku (given an image of a incomplete Sudoku board, the goal is to provide a complete symbolic solution) and Predicate Classification (*PredCl*, given in input labeled and localized bounding boxes of a set of objects in an image, the goal is to predict the correct predicate between them). **(1) Visual Sudoku.** We first compared NASR with different baselines: a *Symbolic Baseline* that executes a Symbolic-Solver after converting the input image to a symbolic form; two state-of-the-art neuro-symbolic methods: *NeurASP* [2] and *SatNet* [3]; and the integration of the latter into NASR (*SatNet+NASR*). The results show that we generally outperform all the other methods and that SatNet performance can be improved (sometimes substantially) by injecting hard constraints via the integration with NASR. We also proved that NASR is more robust to noise compared to the Symbolic-Baseline and established that NASR is the most efficient method in terms of computational time vs performance when compared to the other approaches. **(2) PredCl.** We tested NASR on the GQA dataset (a balanced version of Visual Genome), considering a simple domain-range ontology. We compare against a purely neural baseline [4] and considered the percentage of the max achievable improvement defined by an intractable baseline, consisting of running a probabilistic symbolic solver directly on the output of the neural baseline model. The results show that NASR achieves good performance, and is able to recover the majority of the recoverable errors, leading to a improvement between 1% and 2%.

To conclude, we presented a neuro-symbolic method that aims to efficiently satisfy domain-knowledge constraints at inference. This enables a favourable trade-off between accurate predictions, noise robustness, and computation cost. Our framework is generic and can be applied to different types of input (image, text, etc.) and constraints type (logic, arithmetic, etc.).

References

[1] C. Cornelio et al. Learning where and when to reason in neuro-symbolic inference, In *Proc. of The 11th International Conference on Learning Representations (ICLR)*, 2023.
[2] Z. Yang et al. NeurASP: Embracing neural networks into answer set programming. IJCAI, 2020.
[3] P.W. Wang et al. Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. ICML, 2019.
[4] B. Knyazev et al. Graph density-aware losses for novel compositions in scene graph generation. BMVC, 2020.