Property Invariant Neural Network for Embedding Formulas in CNF*

Miroslav Olšák[1], Cezary Kaliszyk[1], and Josef Urban[2]

[1] University of Innsbruck, Innsbruck, Austria
[2] Czech Technical University in Prague, Czechia

Abstract

Most approaches for embedding formulas use learned embeddings for the predefined function symbols and constants. That is however not well suited for handling new definitions such as Skolem constants. We present a network that does not take into account global names of functions, considering names always local inside a single query. This approach is suitable for the classification of enough clauses at the same time so that the network can infer the features of the symbols only from their usage.

1 Architecture Outline

The network receives a set of clauses and outputs an embedding vector for every clause, every function and relational symbol, and every literal and every subterm occurring in the clauses. To produce them, the input set of clauses is first encoded into a graph in which every object of the three types above is represented by s node. Od particular interest is the type (3) of subterms and literals since we use perfect sharing among the subterms, that is, if two terms or literals are identical (e.g. same variables), they are represented by a single node.

The graph is provided with two types of edges. There are binary edges between clauses and literals describing what literals belong to what clauses. The second type of edges is 4-ary, containing one symbol node and three term nodes. These edges represent the structure of the terms including the argument orders of functions.

We initialize the graph based on basic node properties only – the origin of the clause, whether a term is a variable, etc. and then we perform several (constant number) message-passing layers on the graph. The output of the network is the output of the last message-passing layer.

By design of the graph, the network is invariant under symbol renaming, reordering the clauses, or reordering the literals in a clause. The invariance under negation is achieved by carefully handling the embeddings of symbols. If a vector (embedding) e represents a relational symbol R, then $-e$ represents the relational symbol $\neg R$. The message passing is designed so that this property is preserved through the layers.

2 Experiments

We have conducted three experiments with the network. The first one is for guiding a tableaux connection prover leanCoP [3], in a reinforcement learning setup similar to rlCoP [2]. The prover obtains a set of clauses on the input, and it tries to form a case analysis tree that would prove a contradiction. The network obtains all the axioms from the input (that do not change), and

*MO and CK were supported by the ERC Project *SMART* Starting Grant no. 714034. JU was funded by the *AI4REASON* ERC Consolidator grant nr. 649043, the Czech project AI&Reasoning CZ.02.1.01/0.0/0.0/15_003/0000466 and the European Regional Development Fund.

also the current tree, and outputs an evaluation of the current state (value) and probability distribution (policy) of available actions; the actions are represented by literals in the axioms. First, we ran leanCoP with random guidance on Miz40 dataset to get training data, then, we trained the network's value and policy on a training part of the solved problems (90% of them), and then we ran leanCoP guided by Monte Carlo Tree Search (MCTS) based on the network's estimation. In the MCTS we expand each node 200 times before making a bigstep, and the prover has a limit of 200 steps to prove the theorem.

The random prover solved 4595 training and 510 testing problems. The MCTS prover with network guidance then solved 11978 training and 1322 testing problems in the first iteration and 12648 training and 1394 in the second one. This does not reach the results of the original rlCoP, however, it seems to be rather caused by stricter limits despite better predictions.

Our second experiment is premise selection on DeepMath dataset [1], in this experiment, we consider a whole premise with all its candidate premises as a single query and the network can, therefore, use the other premise candidates to determine whether a particular candidate is positive or negative. The network achieved around 80% testing accuracy on this task.

The third experiment is also performed on DeepMath dataset with an atypical objective. Since our network uses only the structure of the formulae and returns also embeddings of the symbols, we trained it to predict the symbol names in the formula and then tested it on the testing part of the dataset. The network achieved around 65% testing accuracy on this task.

3 Related work

Graph Neural Networks for formula embedding invariant under variable renaming were previously used in the FormulaNet [5] mainly for experiments with higher order logic. A different invariance property was proposed in a network for propositional calculus in the NeuroSat [4]. This network is invariant under negation, order of clauses, and order of literals in clauses, however, this is restricted to propositional logic, where no quantifiers and variables are present.

References

[1] Alexander A. Alemi, François Chollet, Niklas Eén, Geoffrey Irving, Christian Szegedy, and Josef Urban. DeepMath - deep sequence models for premise selection. In Daniel D. Lee, Masashi Sugiyama, Ulrike V. Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2235–2243, 2016.

[2] Cezary Kaliszyk, Josef Urban, Henryk Michalewski, and Miroslav Olšák. Reinforcement learning of theorem proving. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 8836–8847, 2018.

[3] Jens Otten and Wolfgang Bibel. leanCoP: lean connection-based theorem proving. *J. Symb. Comput.*, 36(1-2):139–161, 2003.

[4] Daniel Selsam, Matthew Lamm, Benedikt Bünz, Percy Liang, Leonardo de Moura, and David L. Dill. Learning a SAT solver from single-bit supervision. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[5] Mingzhe Wang, Yihe Tang, Jian Wang, and Jia Deng. Premise selection for theorem proving by deep graph embedding. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information*

Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, pages 2783–2793, 2017.