Training Neural Networks to Do Logic, with Logic

Paul Tarau

Department of Computer Science and Engineering
University of North Texas
paul.tarau@unt.edu

We describe combinatorial generation algorithms, with focus on lambda terms and related type inference algorithms, all elegantly expressible in a logic programming language that supports backtracking and unification.

With help from these tools, we introduce methods to train neural networks as theorem provers. Our combinatorial generation algorithms provide pairs of lambda terms together with their inferred types. We make use of the Curry-Howard isomorphism between lambda terms and formulas in linear and intuitionistic logic corresponding to their types to train our neural networks on large, combinatorially generated datasets mapping formulas to their proof terms.

The dataset, containing generated theorems of several sizes and their proof-terms in postfix form, is available at `http://www.cse.unt.edu/~tarau/datasets/lltaut/` and can be used for correctness, performance and scalability testing for linear logic theorem provers. Our experiments with training Recurrent Neural Networks using our implicational linear logic theorem dataset are available at: `https://github.com/ptarau/neuralgs`.

Keywords: logic programming tools for theorem proving, intuitionistic and linear logic, Curry-Howard isomorphism, neural theorem proving, neuro-symbolic computing