POSUDEK VEDOUCÍHO NA DIPLOMOVOU PRÁCI MARTIN SMOLÍK: NEURAL MODELLING OF MATHEMATICAL STRUCTURES AND THEIR EXTENSIONS

The field of Automated Theorem Proving (ATP), also called Automated Deduction (AD), stands out of the variety of other scientific fields because of its basic relevance for virtually any other scientific discipline. In particular, to appreciate this universal and cross-disciplinary character just recall the nature of mathematics, which is undoubtedly a basis for hard sciences.

ATP as the field of automating mathematics or math for short naturally inherits this fundamental relevance of math. Math is known as one of the hardest disciplines for the human mind. ATP as the meta-science of math, ie. the science which tries to reveal the mechanisms underlying creative mathematical achievements, might thus be judged as even harder than math, since doing seems generally easier than understanding the doing. This extreme hardness of ATP explains why, despite of more than 60 years of ATP research and despite steady progress and impressive singular achievements, there is still a long way to go until ATP and AD can be said to match the level of top mathematicians or, more generally, of scientists in mathematical or deductive reasoning.

Most of the existing ATP approaches are so far based on relatively simple syntactic methods. They represent objects of mathematical reasoning as syntactic trees (terms, formulas, proofs) and guide automated proof search based on syntactic characterizations of such objects. This may be insufficient to capture high-level mathematical reasoning that often concerns typical mathematical structures (models) and the validity of particular claims and properties in them. For example, one may quickly come up with the group S3 as a counterexample to the claim that all groups are abelian, rather than toiling with syntactic proofs and disproofs of such a claim. And one may quickly come up with integers as a counterexample to the claim that all finitely generated groups are finite.

It seems that trained mathematicians gradually develop a set of useful examples and counterexamples like these and use them for fast initial evaluation of conjectures in their theories. Can we simulate such semantic reasoning methods on computers? We would first need to effectively represent important math structures and their properties in computers. This is a hard task: the set of real numbers is a common example, but it does not fit fully in today's computers. Yet, despite the finiteness of their brains, mathematicians still develop useful intuition about real numbers and integers. Likely, some form of (organic) neural networks are responsible for this in the trained brains.

The submitted thesis "Neural modelling of mathematical structures and their extensions" makes the first experiments to automatically train approximations of mathematical structures as neural networks. The thesis focuses on small cyclic and permutation groups, but it is clear that this can be generalized to arbitrary math structures. The networks can in principle be trained only from (some) properties that hold about a particular structure, avoiding the necessity of exhaustive representation of large and infinite objects in computer memory. The working assumption is that this is what trained mathematicians may often do too.

While this is clearly just the initial step in this research program, the first results are interesting. The thesis shows that training small cyclic groups is relatively straightforward, while permutation groups are harder to learn by standard differentiable methods such as backpropagation. This poses a further natural research question: to what extent do humans use backpropagation to gain intuition about complicated math structures? And in what representation space is the learning done? The thesis already shows that the choice of representation may matter a lot for the learning.

Another interesting aspect the thesis studies is the capability of the learned representations to capture extensions. It is shown that in cyclic groups, discovering the interpretation of a "half" element works quite reasonably, while it seems harder for permutation groups. These first experiments again pave the way to further experiments with training arbitrary extensions of arbitrary structures as another important aspect of mathematicians' research.

As usual in topics that bridge multiple scientific fields, the thesis requires the readers to dive into areas that may be out of their zone of comfort. My longtime experience is that the two communities, ie. (formal) mathematicians and machine learning researchers, are used to different level of detail and formality, and that they consider different things as (non)obvious and (un)worthy of explanation. Based on what I have seen so far, I believe that the thesis does a relatively good job in trying to make things accessible to both communities by including relatively detailed background sections on model theory and neural networks. But I am sure the thesis will still not be easily accessible to many people from both communities.

Finally, I commend the author for his courage to undertake such a dangerous and challenging topic, and sufficiently learning the explosively growing field of neural networks, both theoretically and practically. Getting down to programming computers to do reasoning is a very nontrivial task, and there have been so far only a handful of mathematicians that have endured that. It requires learning quite different skills than those mastered during mathematical training, and frankly, it may be very annoying and intimidatingly slow compared to the speed, lightness and pleasure of doing math in one's head. The more I am pleased by the fact that the work resulted in findings that seem actually already interesting, and that the project was judged interesting enough by the reviewers to be presented at the AITP'19 conference.