Automated Theorem Provers as the Hub of the AI Math Ecosystem

Stephan Schulz

DHBW Stuttgart, Stuttgart, Germany
schulz@eprover.org

Keywords: Automated reasoning · Formal mathematics · Artificial intelligence.

1 AI and Mathematics

Modern AI has a history going back about three quarters of a century. The field can be broadly split into symbolic and sub-symbolic approaches on the one hand, and into reasoning vs. learning on the other hand. The core distinction between symbolic and sub-symbolic systems is that symbolic systems represent knowledge as collections of specific, discrete, often logic-based objects, while sub-symbolic systems represent knowledge distributed in a large, usually numerical set of parameters.

Both symbolic and sub-symbolic approaches have produced impressive machine learning methods. On the symbolic side we find e.g. methods based on decision trees, similarity and analogy, and evolutionary methods. On the sub-symbolic side, the most prominent approach is that of artificial neural networks. These typically consist of layers of simple neurons that are interconnected and pass numerical values from one layer to the next to produce an output vector from an input vector. They are trained via error back-propagation. These techniques have been known since the 1970s, but in recent years the combination of much more powerful hardware, much larger datasets, and some changes in network layout and activation functions has inspired and enabled the development of *deep neural networks*. These typically have many more layers than older systems, they have a more complex architecture, interleaving fully connected layers, convolution layers, pooling layers and others. Another big change is that they typically work on much more raw input data and rely less on predefined features. The enormous success of these deep neural networks has fueled the recent successes of artificial intelligence.

In the field of mathematics, symbolic systems have a long history. In particular automated theorem provers (ATP) and computer algebra systems use sound symbolic inference to perform reasoning steps and provide derivations [13] that are correct by construction (at least in the ideal case) and that can be verified a-posteriori to provide extremely reliable results. However, these systems have weaknesses. First, knowledge acquisition is largely based on manual encoding of mathematical theories - a process that is expensive and error-prone. Secondly, the space of possible logical derivations is to large that it is hard to find the interesting reasoning steps, in particular with respect to the task of proving a given conjecture.

On the sub-symbolic side, the combination of deep neural networks with word vector encodings has enabled the creation of large language models (LLMs) - basically neural networks that learn the conditional probability of the next word (more precisely *token*) of a text based on a given context constructed both from an initial prompt and the text generated so far. Such models, trained on internet-sized datasets and applied recursively to their own output, have demonstrated extremely impressive capabilities. They are able to routinely perform language tasks such as summarizing, translation, and reformulation, and they appear as quite intelligent conversation partners. In particular, they apparently can solve mathematical and logical puzzles, which hase lead to the idea that such systems might also be useful formal mathematical reasoning.

However, despite their success in conversational settings, I believe that plain large language models will be inherently unable to reliably generate new mathematical results - essentially because they learn models of language and text, not models of real or mathematical structures. As such, they are limited to reproduce existing ideas (even if in many variations) over existing structures. A large part of mathematics, on the other hand, is the discovery of abstract properties of new structures. While LLMs have been able to apparently solve many logical puzzles, the success seems to drastically drop off if such puzzles are reworded with new vocabulary, or if confounding variables are added [6]. Moreover, LLMs tend to *hallucinate*, basically producing intelligent looking but completely counterfactual or nonsensical text streams. Such hallucinations may be unavoidable [1].

2 Hybrid Architectures for Mathematical AI

I believe that scalable, usable mathematical AI systems must be based on hybrid architectures. The core collection of mathematical knowledge will be encoded in a formal logic, most likely variants of higher-order predicate logic with large first-order subsets. The integrity of this knowledge base will be supported by automated theorem provers such as E [12, 11, 16, 17] and Vampire [5], which will both help to maintain the consistency of this knowledge as new domains are added (as e.g. in [14]), and provide ways to derive new theorems and flesh out new theories. Interactive systems such as e.g. Lean [7] 1or Isabelle [8] will provide the user interface for human mathematicians.

Various machine learning method and classical optimisation approaches will help ATPs to deal with the complexities of the search space, as already demonstrated by several systems [10, 2, 4, 3, 15, 9].

I see the role of LLM-based approaches not in executing actual reasoning, but as tools to facilitate the formalisation of existing mathematical literature and to support the generation of human-readable proofs.

References

1. Banerjee, S., Agarwal, A., Singla, S.: LLMs Will Always Hallucinate, and We Need to Live With This (2024), https://arxiv.org/abs/2409.05746
2. Chvalovský, K., Jakubuv, J., Suda, M., Urban, J.: ENIGMA-NG: Efficient Neural and Gradient-Boosted Inference Guidance for E. In: Fontaine, P. (ed.) Proc. of the 27th CADE, Natal, Brasil. pp. 197–215. No. 11716 in LNAI, Springer (2019)
3. Goertzel, Z.A., Chvalovský, K., Jakubův, J., Olšák, M., Urban, J.: Fast and Slow Enigmas and Parental Guidance. In: Konev, B., Reger, G. (eds.) 13th International Symposium on Frontiers of Combining Systems. pp. 173–191. Springer (2021)
4. Jan Jakubuv, K.C., Olsák, M., Piotrowski, B., Suda, M., Urban, J.: ENIGMA Anonymous: Symbol-Independent Inference Guiding Machine (System Description. In: Peltier, N., Sofronie-Stokkermans, V. (eds.) Proc. of the 10th IJCAR, Paris (Part II). LNAI, vol. 12167, pp. 448–463. Springer (2020)
5. Kovács, L., Voronkov, A.: First-order theorem proving and Vampire. In: Sharygina, N., Veith, H. (eds.) Proc. of the 25th CAV, LNCS, vol. 8044, pp. 1–35. Springer (2013)
6. Mirzadeh, I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S., Farajtabar, M.: GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models (2024), https://arxiv.org/abs/2410.05229
7. Moura, L.d., Ullrich, S.: The Lean 4 Theorem Prover and Programming Language. In: Platzer, A., Sutcliffe, G. (eds.) Proc. of the 28th CADE, Pittsburgh. pp. 625–635. Springer (2021)
8. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL: A Proof Assistant for Higher-Order Logic, LNCS, vol. 2283. Springer (2002)
9. Schäfer, S., Schulz, S.: Breeding theorem proving heuristics with genetic algorithms. In: Gottlob, G., Sutcliffe, G., Voronkov, A. (eds.) Proc. of the Global Conference on Artificial Intelligence, Tibilisi, Georgia. EPiC, vol. 36, pp. 263–274. EasyChair (2015)
10. Schulz, S.: Learning Search Control Knowledge for Equational Theorem Proving. In: Baader, F., Brewka, G., Eiter, T. (eds.) Proc. of the Joint German/Austrian Conference on Artificial Intelligence (KI-2001). LNAI, vol. 2174, pp. 320–334. Springer (2001)
11. Schulz, S.: E – A Brainiac Theorem Prover. Journal of AI Communications **15**(2/3), 111–126 (2002)
12. Schulz, S., Cruanes, S., Vukmirović, P.: Faster, higher, stronger: E 2.3. In: Fontaine, P. (ed.) Proc. of the 27th CADE, Natal, Brasil. pp. 495–507. No. 11716 in LNAI, Springer (2019)
13. Schulz, S., Sutcliffe, G.: Proof generation for saturating first-order theorem provers. In: Delahaye, D., Woltzenlogel Paleo, B. (eds.) All about Proofs, Proofs for All, Mathematical Logic and Foundations, vol. 55, pp. 45–61. College Publications, London, UK (January 2015)
14. Schulz, S., Sutcliffe, G., Urban, J., Pease, A.: Detecting inconsistencies in large first-order knowledge bases. In: de Moura, L. (ed.) Proc. of the 26th CADE, Gothenburg. LNAI, vol. 10395, pp. 310–325. Springer (2017)
15. Urban, J.: Blistr: The blind strategymaker. In: Gottlob, G., Sutcliffe, G., Voronkov, A. (eds.) Proc. of the Global Conference on Artificial Intelligence, Tibilisi, Georgia. EPiC, vol. 36, pp. 312–319. EasyChair (2015)
16. Vukmirović, P., Blanchette, J.C., Cruanes, S., Schulz, S.: Extending a Brainiac Prover to Lambda-free Higher-Order Logic. International Journal on Software

Tools for Technology Transfer (August 2021). https://doi.org/10.1007/s10009-021-00639-7

17. Vukmirović, P., Blanchette, J.C., Schulz, S.: Extending a high-performance prover to higher-order logic. In: Sharygina, N., Sankaranarayanan, S. (eds.) Proc. 29th Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'23), Paris, France. pp. 111–132. No. 13994(2) in LNCS, Springer (2023)