A Theory for AI Computation?

(Extended Abstract)

Ana Ozaki
University of Oslo, Norway

Artificial Intelligence (AI) has become ubiquitous in our everyday lives. There has been intense research on tasks related to language generation and understanding and, as a result of the improvements in these tasks (see e.g. BERT [5], (chat)GPT [3], and Bard), AI has been applied to enhance user experience in household appliances, virtual personal assistants, search engines, recommender systems, among others. A big portion of articles on machine learning is based on experimental evaluations, where the goal is often to provide evidence of an improvement of accuracy results on benchmarks. There are more and more of such papers being published every year, with authors in the machine learning community going from one deadline to the next every two months [2]. While the advance of application-driven research in AI at such a pace is exciting, we also envision an urgent need of establishing and standardizing a formalization of certain notions used in the field, starting with basic conceptual questions. **What is a learning algorithm**? **What does it mean (formally!) to say that it has learnt to perform a certain task**? Exciting research in AI needs to be complemented with theoretical developments, which aim at long-term impact and can survive the test of time. The lack of formal definitions and their consistent usage in machine learning can create misunderstandings in the community and expectations in the society that AI technologies have achieved capabilities that are later shown to be wrong [14, 22] or unsuitable because of ethical issues [10].

Remark 1 To illustrate the issue regarding the lack of formal definitions and their usage, consider a very intriguing question in AI which is whether algorithms can learn by *induction* the ability of performing *deductive (logical) reasoning*. After multiple authors claimed that the BERT language model learnt to emulate the correct function for performing deductive reasoning [4, 23] (based on experimental evaluations with high accuracy prediction results on benchmarks), further studies indicate that the model learnt *statistical features* present in deductive reasoning problems, rather than learning to emulate the correct reasoning function. Their results point that BERT has *not* learnt to reason [25] and, moreover, that this would not be fixable by feeding the model with more data as the statistical features in this case are *inherent* to the reasoning function itself, and therefore, they would be present in any kind of data distribution.

The presence of *shortcut learning* [6], as illustrated in Remark 1 (where

the model learns statistical features instead of emulating the intended target reasoning function), and its effects are yet to be understood. The solutions may require more interaction between the communities working on Knowledge Representation and Reasoning, Logic, and Machine Learning, to achieve models that can perform both deductive and inductive reasoning adequately.

On a deeper level, what are the **capabilities and limitations of AI**? Back in time, when researchers were trying to understand the capabilities and limitations of computers, they found many answers to their questions by developing what is known today as the **theory of computation** [21]. Research in the field clarifies which problems can be solved by computers, formalizing computers using abstract computational models and analysing the decidability, complexity, and potential reducibility of computational problems. The Theory of Computation provides a solid and elegant foundation for computer science. The well-defined models of computation create a mathematical abstraction that allows the study of capabilities and limitations of computers independently of taking into consideration the hardware resources of a particular computer.

However, the mode of operation of classical algorithms is fairly different from the settings predominant in AI. Systems based on AI often include interactions with external entities, which translates computationally into multiple inputs and outputs during the computation [8]—a much less explored territory in the literature of the theory of computation. Though, the challenges in AI include not only analysing the amount of time and memory space required but also investigating the capability of learning target functions, updating the learnt concepts given new information, estimating the number of interactions needed (in interactive scenarios), and estimating the amount of data needed for training (in non-interactive scenarios) so as to avoid overfitting [20]. There has been a lot of interest in verifying whether AI models can be given formal guarantees, such as robustness to adversarial attacks [12, 16, 19, 9], and establishing the expressivity of widely applied machine learning architectures such as transformers [18] and graph neural networks [15]. In the quest for a clear and solid understanding of the capabilities and limitations of AI, some of the basic but also fundamental questions to be addressed towards establishing a theory for AI computation include: what are the underlying **computational models** of AI systems? How can we formally define **learnability**? What is the **complexity of learning**? Can we **reduce** one learning problem to another?

In Computational Learning Theory [11], there are well-known learning frameworks such as the exact [1] and the classical probably approximately correct (PAC) [24]. Within these frameworks, one can formalize a learning problem, analyse learnability, complexity, reducibility, and under certain conditions, estimate an amount of training data (based on the notion of sample complexity) that avoids overfitting within the model. Although research in Computational Learning Theory produced a plethora of important results for traditional algorithmic tasks, it is greatly underdeveloped when it comes to computational settings and challenges predominant in AI.

Remark 2 To illustrate the mismatch between what is known in theory and

what is required in practice, consider the assumptions made in well studied computational models in Computational Learning Theory. The classical PAC framework makes the assumption that examples in the training data are *independently and identically distributed (i.i.d)* and uses this **same**—arbitrary but **fixed**—distribution to establish the notion of probabilistic approximation w.r.t. the target, the function that is *intended* to be learnt. However, the assumption that the test data comes from the same distribution as the training data has been called **the big lie in machine learning** [7]. The reason is because it is convenient to make this assumption "in the lab", though, this assumption is rarely justified for real world applications and gives more opportunities for (unintended) shortcut learning [6]. In the exact learning framework, the assumption that the learner has access an oracle with perfect knowledge of the target, in particular, an oracle that can provide counterexamples to equivalence queries, is also difficult to fulfill in practice. Moreover, a big portion of works in the field only consider supervised settings based on binary classification.

Starting from basic conceptual questions, what is a learning algorithm when we now consider interactive systems that receive and return multiple inputs and outputs? The theory of computation has a classical notion of algorithm for **decision problems**, based on Turing Machines. There are also branches of the Theory of Computation which resemble scenarios with interaction and changes, as in many AI scenarios. For instance, Oracle Turing machines give the possibility of interacting with an external system [13]. The work by Goldin, Smolka, Wegner et al. provides foundational results for interactive computation [8]. Also, in the study of Dynamic Complexity Classes [17], one has to consider modifications in the data. However, these formalisms are considerably different from the idea of having computational models that can be employed to explore notions associated with **learning tasks** such as learnability, sample complexity, and query complexity (in interactive settings). The idea of having for frameworks with a success criteria based on learning tasks, instead of decision tasks is not within the realms of such formalisms. In this extended abstract, we cast light on abysmal literature gaps between AI and foundational research. We motivate the need of developing a theory for AI computation, that builds on Machine Learning, Knowledge Representation and Logic—the pillars of AI—as well as Computational Learning Theory and the Theory of Computation.

References

[1] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[2] Yoshua Bengio. Time to rethink the publication process in machine learning, 2020.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*. Curran Associates Inc., 2020.

[4] Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. In *IJCAI*, pages 3882–3890. International Joint Conferences on Artificial Intelligence Organization, 2020.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[6] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nat. Mach. Intell.*, 2(11):665–673, 2020.

[7] Z. Ghahramani. Panel of workshop on advances in approximate bayesian inference (AABI), 2017. `https://www.youtube.com/watch?v=x1UByHT6OmQ&feature=youtu.be&t=37m44s`.

[8] Dina Goldin, Scott A. Smolka, and Peter Wegner. *Interactive Computation: The New Paradigm*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[9] Kai Jia and Martin Rinard. Efficient exact verification of binarized neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1782–1795. Curran Associates, Inc., 2020.

[10] Michael Kearns and Aaron Roth. *The Ethical Algorithm: The Science of Socially Aware Algorithm Design*. Oxford University Press, Inc., 2019.

[11] Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

[12] Emanuele La Malfa and Marta Kwiatkowska. The king is naked: On the notion of robustness for natural language processing. In *AAAI*, pages 11047–11057, 2022.

[13] Dieter Melkebeek. *Randomness and Completeness in Computational Complexity*. Springer, 2000.

[14] Rachel Metz. Google fires engineer who contended its ai technology was sentient, 2022.

[15] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *AAAI*, pages 4602–4609, 2019.

[16] Nina Narodytska, Shiva Prasad Kasiviswanathan, Leonid Ryzhyk, Mooly Sagiv, and Toby Walsh. Verifying properties of binarized deep neural networks. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *AAAI*, pages 6615–6624, 2018.

[17] Sushant Patnaik and Neil Immerman. Dyn-fo: A parallel, dynamic complexity class. *Journal of Computer and System Sciences*, 55(2):199–209, 1997.

[18] Jorge Pérez, Javier Marinkovic, and Pablo Barceló. On the turing completeness of modern neural network architectures. In *ICLR*, 2019.

[19] Emilia Przybysz, Bimal Bhattarai, Cosimo Persia, Ana Ozaki, Ole-Christoffer Granmo, and Jivitesh Sharma. Verifying properties of tsetlin machines. *CoRR*, abs/2303.14464, 2023. (to appear) ISTM.

[20] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[21] Michael Sipser. *Introduction to the Theory of Computation*. Thomson Course Technology, international edition of second edition, 2005.

[22] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Trans. Evol. Comput.*, 23(5):828–841, 2019.

[23] Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. In *NeurIPS*. Curran Associates Inc., 2020.

[24] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

[25] Honghua Zhang, Liunian Harold Li, Tao Meng, Kai-Wei Chang, and Guy Van den Broeck. On the paradox of learning to reason from data. *CoRR*, abs/2205.11502, 2022.