

# CONNECTION OF MACHINE LEARNING AND MATHEMATICS

*Serhii Krochak*

*Institute for Applied System Analysis,*

*National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”*

The field of Machine Learning (ML) has been thriving in recent years. As a result, it has impacted several other branches of research, including mathematics. Several instances of such interdisciplinary integration will be explored in this thesis.

The primary goal is to conduct research on the history of ML and recent developments of specific models that are tangent to mathematics, AlphaGeometry in particular. Various journals, textbooks, and websites will be used to find relevant information.

ML is a relatively recent field of science, with the term first being mentioned in 1952 by Arthur Samuel, an IBM computer scientist (Small, 2023, p. 1). One of the first prototypes of modern neural networks was created in 1951 by Marvin Lee Minsky; it utilised an arrangement of lights that could encode various phenomena (Stochastic Neural Analog Reinforcement Calculator, 2025). For instance, it was used to solve Shannon’s maze, this fact foreshadowed the potential of such devices in problem-solving. Based on this device, the perceptron, “the first neural network”, was invented in 1958 by Frank Rosenblatt (Lefkowitz, 2019). S. Welch (2025) explains the core principle behind this machine.

Since then, multiple developments, including backpropagation, have been created and repeatedly used to solve mathematical problems. In 1989, G. Cybenko published a paper that established this fact, yielding a powerful result in approximating functions. In their paper, which would further extend the findings, Wang & Qu (2021) stated: “The universal approximation theorem of Cybenko states that single hidden layer neural networks can arbitrarily well approximate any continuous function with support in the unit hypercube” (p.1). This led to Physically-Informed Neural Networks (PINN) being used to solve Partial Differential Equations (PDE) ever since the 1990s. The importance of this step is that non-linear PDEs

usually do not have closed-form solutions, hence only computational solving methods remain available. However, they usually are time-consuming; hence, using ML is of extreme benefit, as not only is it faster, but it can also be more precise. As Beck et al. (2023) mentioned, “Such deep learning-based approximation methods for PDEs have first been proposed in the 1990s in the case of low-dimensional PDEs, cf., e.g., Dissanayake & Phan-Thien” (p. 3698). Up until recent years, solving PDEs was one of the primary purposes of using AI.

Another big leap in terms of neural network development was the introduction of the Transformer architecture by A. Vaswani et al. (2017) in 2017. It was based solely on attention mechanisms, dispensing with recurrence and convolutions entirely, hence being superior in quality while being more parallelizable and requiring significantly less time to train than previous models. Then came the time for researchers to improve and build upon this architecture. J. Kaplan et al. (2020) discovered the scaling laws for neutral language models, which led to bigger models and using more data and, thereafter, increasing their output quality. A month later, A. Roberts et al. (2020) posted a paper, which was one of the first ones to introduce the term “Large Language Model” (LLM), and several months after that, GPT-3 (generative pre-trained transformer) was introduced, setting the path for the LLMs we know today (Brown et al., 2020). Switch transformers (that focus on increasing the parameter count, as opposed to increasing the model size) were invented the following year, together with Mixture-of-Experts (that allows the parameters to vary between inputs), which made the creation of trillion-parameter models possible (Fedus et al. 2021). There were countless incremental developments improving the scalability and efficiency of large models.

However, scalability was not the only restriction that was reduced. For example, Chain-of-Thought prompting was introduced in 2022 (Wei et al., 2022). It was proof that intermedia-step prompts could significantly improve the output quality for LLMs. In the same year, Refinement Learning from Human Feedback was introduced, which would not only increase learning efficiency but also allow for models to be more specific, leading to safer interactions (Ouyang et al., 2022). It also

allows for LLMs to be trained to solve math-specific tasks. Another step in LLM development was Retrieval-augmented generation, which allowed users to receive up-to-date outputs from the models (Lewis et al., 2020).

Finally, a model, “pretrained on general natural language data and further trained on technical content” (Lewkowycz et al., 2022, abs), called “Minerva”, was created. It focused on what other LLMs struggled with – quantitative reasoning, and managed to answer a third of the undergraduate problems it was tested on.

Now that we have a clear understanding of ML history, it is safe to explore some recent math-specific models.

It has already been mentioned that PINNs are usually used for solving PDEs. Currently, they have been scaled to high-dimensional PDEs, as well as to including some unstable, singular solutions (Wang et al., 2025).

Algebraic equalities do not appear as something that ML can be used on, at least because of small dataset sizes. However, the Algebraic Inequality Proving System (AIPS) solved the issue by generating families of inequality problems of increasing difficulty to train its components. The system combines symbolic inequality transformations (inequality-solving methods) with learned guidance that generates proof paths. AIPS also uses curriculum strategies and value networks to prioritise search branches; this is crucial for the huge search space of algebraic manipulations. Wei et al. (2024) noted that: “On a test set of 20 International Mathematical Olympiad-level inequality problems, AIPS successfully solved 10, outperforming state-of-the-art methods. Furthermore, AIPS automatically generated a vast array of non-trivial theorems without human intervention”.

And another popular field of mathematics, geometry, was also majorly affected by ML. The information about AlphaGeometry solving IMO-level problems spread throughout the Internet like wildfire. The core principle behind this model is the same as the previous one’s: AlphaGeometry synthesises millions of geometry problems and proofs to train models; an LLM parses natural language, proposes high-level steps or lemmas; a symbolic geometry engine (formal geometry language) executes steps and verifies correctness; the system integrates neural proposals with exhaustive

symbolic search, yielding correct proofs rather than just plausible text. “On a test set of 30 latest olympiad-level problems, AlphaGeometry solves 25, outperforming the previous best method that only solves ten problems and approaching the performance of an average International Mathematical Olympiad (IMO) gold medallist” (Trinh et al., 2024, abs).

Mathematics has been one of the driving purposes for the development of ML, which, in turn, has greatly contributed to the branches of PDE, algebra, and geometry. Given the trends, it will not be rare to witness a model combining an LLM and a symbolic engine to solve problems.

### References:

1. Small, E. (2023). *An analysis of physics-informed Neural Networks* [Master's dissertation, the University of Manchester]. Arxiv. <https://arxiv.org/pdf/2303.02890#:~:text=Machine%20Learning%20is%20a%20relatively,9%5D%2C%20to%20name%20a>
2. *Stochastic Neural Analog Reinforcement Calculator*. (2025, July 7). Wikipedia. [https://en.wikipedia.org/wiki/Stochastic\\_Neural\\_Analog\\_Reinforcement\\_Calculator#:~:text=The%20%20Stochastic%20Neural%20Analog,5](https://en.wikipedia.org/wiki/Stochastic_Neural_Analog_Reinforcement_Calculator#:~:text=The%20%20Stochastic%20Neural%20Analog,5)
3. Lefkowitz, M. (2019, September 25). *Professor's perceptron paved the way for AI – 60 years too soon*. <https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon#:~:text=%E2%80%9CThe%20perceptron%20was%20the%20first,%E2%80%9D>
4. Welch, S. (2025, February 1). ChatGPT is made from 100 million of these [The Perceptron]. YouTube. <https://youtu.be/l-9ALe3U-Fg?si=ZwDeoiui-IjWfQL5>
5. Wang, M.-X., & Qu, Y. (2021). Approximation capabilities of neural networks on unbounded domains. *Neural Networks*, 145, p. 1. <https://doi.org/10.1016/j.neunet.2021.10.001>

6. Beck, C., Hutzenthaler, M., Jentzen, A., & Kuckuck, B. (2023, June). *An overview on deep learning-based approximation methods for partial differential equations*. <https://www.aimsciences.org/article/doi/10.3934/dcdsb.2022238>
7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023, August 2). *Attention is all you need*. arXiv.org. <https://arxiv.org/abs/1706.03762>
8. Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020, January 23). *Scaling laws for neural language models*. arXiv.org. <https://arxiv.org/abs/2001.08361>
9. Roberts, A., Raffel, C., & Shazeer, N. (2020, October 5). *How much knowledge can you pack into the parameters of a language model?*. arXiv.org. <https://arxiv.org/abs/2002.08910>
10. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020, July 22). *Language models are few-shot learners*. arXiv.org. <https://arxiv.org/abs/2005.14165>
11. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2023, January 10). *Chain-of-thought prompting elicits reasoning in large language models*. arXiv.org. <https://arxiv.org/abs/2201.11903>
12. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022, March 4). *Training language models to follow instructions with human feedback*. arXiv.org. <https://arxiv.org/abs/2203.02155>
13. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2021, April 12). *Retrieval-augmented generation for knowledge-intensive NLP tasks*. arXiv.org. <https://arxiv.org/abs/2005.11401>

14. Wang, Y., Bennani, M., Martens, J., Racanière, S., Blackwell, S., Matthews, A., Nikolov, S., Cao-Labora, G., Park, D. S., Arjovsky, M., Worrall, D., Qin, C., Alet, F., Kozlovskii, B., Tomašev, N., Davies, A., Kohli, P., Buckmaster, T., Georgiev, B., ... Lai, C.-Y. (2025, September 17). *Discovery of unstable singularities*. arXiv.org. <https://arxiv.org/abs/2509.14185>
15. Wei, C., Sun, M., & Wang, W. (2024, October 31). *Proving olympiad algebraic inequalities without human demonstrations*. arXiv.org. <https://arxiv.org/abs/2406.14219>
16. Trinh, T. H., Wu, Y., Le, Q. V., He, H., & Luong, T. (2024, January 17). *Solving olympiad geometry without human demonstrations*. Nature News. <https://www.nature.com/articles/s41586-023-06747-5>