UDC 004.582:004.946

D. O. Borbuliak
V. V. Chopliak

ALGORITHMIC COMPLEXITY AND ADVANCED MATHEMATICS: OPTIMIZING NEURAL NETWORKS THROUGH LOW-LEVEL IMPLEMENTATIONS

Vinnytsia National Technical University

Анотація

Ефективність нейронних мереж безпосередньо залежить від математичної оптимізації та швидкодії обчислювальних алгоритмів на рівні архітектури процесора. Поєднання апарату вищої математики з низькорівневим програмуванням дозволяє мінімізувати обчислювальні витрати при реалізації складних операцій матричного числення та градієнтного спуску. У роботі аналізується, як пряме керування пам'яттю та векторизація обчислень у мовах системного рівня дозволяють значно прискорити навчання глибоких моделей порівняно з високорівневими інтерпретованими мовами. Особлива увага приділяється алгоритмічній складності та методам апроксимації функцій, які вимагають максимальної точності та оптимізації на рівні заліза для забезпечення роботи систем штучного інтелекту в реальному часі.

Ключові слова: алгоритмічна складність, вища математика, нейронні мережі, низькорівнева оптимізація, матричне числення, керування пам'яттю, обчислювальна ефективність, C++.

Abstract

The efficiency of neural networks directly depends on the mathematical optimization and speed of computational algorithms at the processor architecture level. The combination of higher mathematics with low-level programming allows to minimize computational costs when implementing complex matrix arithmetic and gradient descent operations. The paper analyzes how direct memory management and vectorization of calculations in system-level languages can significantly accelerate the training of deep models compared to high-level interpreted languages. Special attention is paid to algorithmic complexity and methods of function approximation, which require maximum accuracy and optimization at the hardware level to ensure the operation of artificial intelligence systems in real time.

Keywords: algorithmic complexity, higher mathematics, neural networks, low-level optimization, matrix arithmetic, memory management, computational efficiency, C++.

Introduction

In the modern landscape of computational science, the efficiency of neural networks is inextricably linked to the underlying programming language and mathematical optimization. While high-level frameworks offer simplicity, they often struggle with the intensive computational demands of deep learning architectures and large-scale data processing. C++ stands out as a premier choice for developing core AI engines because it provides direct hardware access and deterministic execution, which are vital for maintaining high-speed tensor operations without performance bottlenecks.

One of the most significant advantages of using low-level programming in this field is the granular control over memory management and algorithmic complexity. Unlike environments that rely on a Garbage Collector, C++ allows developers to optimize resource allocation manually, ensuring that neural nodes can process complex matrix transformations and gradient descent with minimal latency. This level of optimization is crucial for building scalable infrastructure that remains stable even under extreme training loads or real-time inference during complex mathematical simulations.

The integration of C++ within machine learning architecture ensures that the system remains both robust and high-performing. This technical synergy between advanced calculus and low-level implementation is the key factor in the successful deployment of next-generation artificial intelligence applications.

Research results

The analysis of neural network architecture reveals that C++ provides a significant performance advantage due to its lack of a virtual machine layer. Unlike many modern frameworks, C++ compiles directly into machine code, which eliminates the execution overhead found in managed environments. This efficiency is particularly visible in matrix multiplications and convolutional operations, where C++ execution speeds consistently outperform higher-level alternatives like Python-based interpreters.

Furthermore, the research highlights the importance of deterministic memory allocation in maintaining computational stability. In deep learning systems, every layer must perform the exact same tensor transformations to ensure gradient consistency. The manual memory management features of C++, such as pointers and custom allocators, prevent the unpredictable pauses caused by automatic garbage collection. This ensures that the time taken to train a single epoch remains consistent across different hardware configurations.

Another key finding is the extensive ecosystem of low-level libraries that facilitate high-performance mathematical development. Most industry-standard optimization tools, including those used for hardware acceleration and SIMD processing, are optimized for C++ to ensure maximum throughput. The ability to integrate these libraries seamlessly allows developers to build highly complex neural models that can handle large data structures without compromising on safety or processing power.

Finally, the study demonstrates that the scalability of artificial intelligence models is deeply dependent on the "close-to-metal" nature of the programming language. By reducing the CPU cycles required for each backpropagation step, C++ enables the network to support a higher number of parameters and more frequent weight updates. This makes it an indispensable tool for developing professional-grade machine learning solutions that require both high throughput and extreme reliability.

Conclusion

In conclusion, the development of robust and efficient neural networks relies heavily on the technical capabilities provided by low-level programming and advanced mathematical optimization. C++ remains an unparalleled tool in the artificial intelligence industry, offering the necessary performance, deterministic execution, and direct memory control that high-level languages often lack. As the digital landscape shifts towards more complex machine learning architectures, the synergy between C++ and algorithmic complexity will continue to be a fundamental requirement for building scalable and reliable intelligent infrastructures.

REFERENCES

1. Stroustrup, B. (2013). The C++ Programming Language (4th ed.). Addison-Wesley Professional. URL: https://chenweixiang.github.io/docs/The_C++_Programming_Language_4th_Edition_Bjarne_Stroustrup.pdf.
2. Heaton, J. Goodfellow, I., Bengio, Y., Courville, A. (2018). Deep learning. *Genet Program Evolvable Mach* 19, 305–307. DOI: https://doi.org/10.1007/s10710-017-9314-z.
3. Knuth, D. E. (1997). The Art of Computer Programming, Volume 1: Fundamental Algorithms. Addison-Wesley. URL: https://seriouscomputerist.atariverse.com/media/pdf/book/Art%20of%20Computer%20Programming%20-%20Volume%201%20(Fundamental%20Algorithms).pdf.
4. Gereron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media. URL: https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632.
5. Kravchenko, K., Ketsyk-Zinchenko, U., Suduk, I., Nykyporets, S., & Cherednychenko, V. (2025). Effectiveness of online platforms in developing language skills of higher education students. *Revista Eduweb*, 19(3), 303-314. DOI: https://doi.org/10.46502/issn.1856-7576/2025.19.03.19.
6. Sachaniuk-Kavets'ka, N. V., & Nykyporets, S. S. (2026). LLM-based automation for translating mathematical formulae and symbols: Challenges and perspectives for technical communication. *Scientific Innovations and Advanced Technologies. Series "Education/Pedagogy"*, 3(55), 660-677. DOI: https://doi.org/10.52058/2786-5274-2026-3(55)-660-677.
7. Sachaniuk-Kavets'ka, N. V., & Nykyporets, S. S. (2025). Developing critical thinking in students of technical specialties through the mathematics of uncertainty and educational debates in English: An integrated experimental-methodological model. *Bulletin of Science and Education. Series "Pedagogy"*, 11(41), 1524-1541. DOI: https://doi.org/10.52058/2786-6165-2025-11(41)-1524-1541.

Борбуляк Дмитро Олександрович – студент групи 5ПІ-25б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: cryptodmytriyme@gmail.com

Науковий керівник: ***Чопляк Вікторія Володимирівна*** – викладач англійської мови, кафедра іноземних мов, Вінницький національний технічний університет, м. Вінниця, e-mail: nikavnuchkova@gmail.com.

Dmytro O. Borbuliak – a student of 5SE-25b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: cryptodmytriyme@gmail.com.

Scientific Supervisor***: Victoriia V. Chopliak*** – teacher of English, Foreign Languages Department, Vinnytsia National Technical University, Vinnytsia, e-mail: nikavnuchkova@gmail.com.