Advances in Computer and Communication, 2025, 6(4), 195-199
https://www.hillpublisher.com/journals/acc/
ISSN Online: 2767-2875

Research on the Optimization Application of Artificial Intelligence Algorithm in Computational Mathematics

Shanshan Xu

Xinjiang Hetian College, Hetian 848000, Xinjiang, China.

How to cite this paper: Shanshan Xu. (2025) Research on the Optimization Application of Artificial Intelligence Algorithm in Computational Mathematics. *Advances in Computer and Communication*, 6(4), 195-199.
DOI: 10.26855/acc.2025.10.007

Received: August 18, 2025
Accepted: September 15, 2025
Published: October 10, 2025

***Corresponding author:** Shanshan Xu, Xinjiang Hetian College, Hetian 848000, Xinjiang, China.

Abstract

The performance and effectiveness of artificial intelligence algorithms are often determined by their level of optimization. Computational mathematics, as a discipline focused on numerical solutions to mathematical problems, provides powerful tools and methodologies for optimizing AI algorithms. By applying the theories and methods of computational mathematics to AI algorithm optimization, we can enhance algorithm efficiency, accuracy, and stability, thereby advancing the development of artificial intelligence technology. This paper explores the critical applications of computational mathematics in AI algorithm optimization, aiming to provide valuable references for better understanding and implementing computational mathematics-driven optimization strategies in AI algorithms.

Keywords

Computational mathematics; Artificial intelligence; Algorithm optimization

Computational mathematics provides a solid theoretical foundation for artificial intelligence algorithms, enhancing the scientific rigor and precision in their design and analysis. In machine learning, this discipline plays a vital role in optimizing algorithm parameters to improve model accuracy and efficiency. It also significantly reduces algorithm complexity while boosting computational speed, enabling AI systems to handle large-scale data and complex tasks. By offering innovative approaches and tools for algorithm refinement, computational mathematics drives continuous progress in artificial intelligence development.

1. Theoretical Basis of Computational Mathematics in Artificial Intelligence Algorithm Optimization

Linear algebra forms the bedrock of artificial intelligence algorithms, particularly in machine learning and deep learning. Data is typically represented as vectors and matrices, where operations like matrix multiplication, transposition, and inversion play pivotal roles in algorithm execution. For instance, weight matrix updates and linear transformations in neural networks fundamentally depend on these matrix computations. Additionally, eigenvalues and eigenvectors are utilized in dimensionality reduction techniques such as principal component analysis (PCA), which help extract key data features while reducing computational complexity [1].

Probability theory and statistics provide the theoretical framework for handling uncertainty and randomness in data. Many machine learning algorithms, such as Naive Bayes classifiers and Hidden Markov Models (HMMs), are based on probabilistic models. Statistical methods like hypothesis testing and confidence intervals help evaluate model performance and reliability. Furthermore, probability distributions and Bayesian inference play crucial roles in algorithms like Bayesian optimization, enabling optimal decision-making even under uncertain conditions.

Calculus plays a vital role in optimization problems, particularly in deep learning. Gradients, a fundamental concept in calculus, guide the direction of parameter updates. The backpropagation algorithm employs the chain rule to compute gradients, thereby optimizing neural network weights. Optimization theories such as gradient descent, Newton's method, and quasi-Newton methods provide effective strategies for minimizing loss functions, enabling models to better approximate data patterns [2].

2. Specific Application of Computational Mathematics in Artificial Intelligence Algorithm Optimization

2.1 Machine learning algorithm optimization

In the field of machine learning, computational mathematics provides a powerful foundation for algorithm optimization. Support Vector Machines (SVM) optimization stands as a prime example, with its core mechanism involving solving convex quadratic programming problems. By transforming classification tasks into such optimization problems, it identifies the optimal hyperplane for categorization. The Sequential Minimal Optimization (SMO) algorithm demonstrates exceptional performance in SVM optimization, breaking down complex quadratic programming into a series of simplified subproblems. This approach optimizes only two variables at a time, significantly reducing computational complexity. As a result, SVM training becomes more efficient, enabling rapid processing of large-scale datasets and enhancing performance in scenarios such as text classification and image recognition. With precision. The optimization of decision tree algorithms also relies on computational mathematics. Information entropy and the Gini coefficient serve as key metrics for evaluating feature importance: information entropy reflects data uncertainty, while the Gini coefficient indicates data purity. These metrics enable effective feature selection and optimization. By selecting crucial features to construct decision trees, we can reduce tree depth and complexity, improve training efficiency, prevent overfitting, and enhance the model's generalization capability [3].

In the feature selection process, computational mathematics methods precisely evaluate feature values to identify the most influential subsets of features for target variables. Beyond Support Vector Machines (SVM) and decision trees, other machine learning algorithms also benefit from computational mathematics. Random forests, composed of multiple decision trees, leverage computational mathematics to optimize their generation and integration processes. By strategically setting parameters and refining feature selection strategies, these methods enhance prediction accuracy and stability. In the K-nearest neighbors algorithm, computational mathematics is applied to refine distance measurement methods and neighbor selection strategies. Accurate distance metrics better assess sample similarity, while appropriate neighbor selection strategies improve classification or regression outcomes. For example, to improve the distance calculation formula by mathematical methods, it is necessary to consider the different feature weights and data distribution characteristics, and optimize the neighbor selection by weighted voting, so that the K nearest neighbor algorithm can play a better role in data mining, recommendation systems, and other fields [4].

2.2 Deep learning algorithm optimization

In neural network training optimization, gradient descent methods and their variants play a crucial role. Gradient descent calculates gradients of loss functions with respect to model parameters, updating them in the direction opposite to the gradient to minimize losses. Stochastic gradient descent (SGD) and its variations demonstrate significant effectiveness in accelerating convergence and enhancing stability. The backpropagation algorithm serves as the core mechanism in neural network training, efficiently calculating gradients through the chain rule by propagating error information from the output layer to the input layer, enabling precise parameter adjustments. In convolutional neural network (CNN) optimization, convolution operations and pooling operations are critical. Convolution operations achieve this through... Convolutional neural networks (CNNs) employ sliding kernels to extract local features by leveraging shared parameters and sparse connections, which significantly reduces the number of parameters. Pooling operations like maximum pooling and average pooling reduce data dimensions while preserving key features, thereby enhancing model robustness. The dual objectives of feature extraction and parameter reduction constitute the core of CNN optimization. Through carefully designed convolutional layers and pooling mechanisms, these networks effectively capture deep-level features from images, audio, and other data types, simultaneously reducing model complexity to improve training efficiency and generalization capabilities. Although recurrent neural networks (RNNs) excel at processing sequential data, they face challenges such as gradient vanishing and gradient explosion. The Long Short-Term Memory network (LSTM) and Gate Recurrent Unit (GRU) are effective solutions to these challenges.

The LSTM employs three gate structures—cell states, input gate, forgetting gate, and output gate—to selectively retain or discard information, effectively capturing long-term dependencies. The GRU simplifies the LSTM by controlling information flow through update gates and reset gates, maintaining performance while reducing computational complexity. The application of computational mathematics in deep learning algorithm optimization extends beyond these specific methods, permeating all stages of model design, training, and evaluation. For instance, through... The network structure is optimized by mathematical analysis, and the appropriate activation function and loss function are selected. The regularization method is used to prevent overfitting [5, 6].

2.3 Reinforcement learning algorithm optimization

In reinforcement learning, the optimization of value functions and policy functions constitutes the core challenges. The dynamic programming approach provides a theoretical foundation for value function optimization by decomposing problems into subproblems and iteratively solving for optimal value functions using Bellman equations. In policy evaluation, dynamic programming enables accurate calculation of value functions under given policies, while in policy iteration, strategies can be continuously optimized based on value functions until the optimal strategy is found. The Monte Carlo method estimates value functions through sampling and averaging. In practical applications, appropriate methods can be selected according to environmental characteristics and problem requirements. For scenarios with known dynamic environmental characteristics, dynamic programming can provide precise solutions, whereas... When the environmental dynamic is unknown, the Monte Carlo method can be used to estimate by sampling data. Deep reinforcement learning combines deep learning with reinforcement learning to further expand the application scope of reinforcement learning [7].

Deep Q Networks (DQNs) represent a typical approach in this field. They utilize deep neural networks to approximate the Q-function, employing an experience replay mechanism that breaks correlations between data points and enhances learning stability. The objective network stabilizes learning objectives while reducing fluctuations during training. Policy gradient methods directly optimize policy parameters from a strategic perspective to maximize expected rewards, eliminating the need for explicit value function estimation through direct optimization of strategies. These methods demonstrate significant advantages in continuous action spaces and high-dimensional state environments, particularly in robotic control tasks where they excel at effectively learning complex control strategies [8].

3. Challenges and solutions of computational mathematics in artificial intelligence algorithm optimization

3.1 High-dimensional data processing

As data dimensions increase, computational complexity and storage requirements grow exponentially. High-dimensional data not only consumes computing resources but also significantly prolongs model training time, potentially leading to overfitting that weakens generalization capabilities. Moreover, distance measurements in high-dimensional spaces become unreliable, making it difficult to accurately assess similarity between data points—a critical factor affecting both prediction accuracy and algorithmic stability [9].

To address these challenges, dimensionality reduction techniques employ mathematical transformations to map high-dimensional data into lower-dimensional spaces while preserving key features and information. Principal Component Analysis (PCA), a widely used linear approach, projects data into new dimensions by identifying principal components, effectively reducing dimensionality. This method not only decreases computational complexity but also enhances model interpretability. t-SNE (t-distribution stochastic neighborhood embedding), a nonlinear technique, is particularly effective for handling complex nonlinear data structures. It achieves this by maintaining data points in both high-dimensional and low-dimensional spaces. By leveraging local similarity within datasets, we achieve effective data visualization. The application of dimensionality reduction techniques not only enhances algorithmic computational efficiency but also strengthens model robustness and generalization capabilities. By reducing data dimensions, models can more effectively capture key features while minimizing the impact of noise and redundant information. Furthermore, the dimensionally reduced data demonstrates significant advantages in visualization and interpretability, making data analysis and model debugging more intuitive and convenient.

3.2 Convergence of the optimization algorithm

In artificial intelligence algorithm optimization, the convergence of optimization algorithms poses a core challenge. Particularly when dealing with complex non-convex optimization problems, algorithms often get trapped in local optima, making it difficult to find global solutions. This significantly impacts model performance and generalization capabilities. The problem of local optima occurs when algorithms discover only partial minima rather than global ones during optimization, preventing models from achieving optimal performance. Furthermore, issues like gradient vanishing and gradient explosion complicate convergence, especially in deep learning models where these challenges are particularly pronounced [10].

To address these challenges, computational mathematics offers multiple effective solutions. The first approach involves introducing randomness, with stochastic gradient descent (SGD) and its variants like small-batch gradient descent being common methods. By randomly selecting data samples for gradient calculations, these approaches enhance the optimization process's randomness, helping algorithms escape local optima while improving convergence speed and global search capabilities. Additionally, momentum methods and Nesterov's accelerated gradient update incorporate momentum terms to accelerate convergence in flat regions and suppress oscillations, thereby further enhancing optimization efficiency. Second-order optimization methods constitute another powerful solution framework. Unlike first-order methods that solely utilize gradient information, second-order methods employ Hessian matrices to capture the second-order derivative information of the objective function, providing more precise gradient direction and step size adjustments. Furthermore, adaptive optimization algorithms can enhance the stability and convergence speed of the optimization process by dynamically adjusting learning rates while combining first-order and second-order moment estimation. These algorithms are widely used in training deep neural networks, effectively addressing issues like gradient vanishing and explosion, thereby improving both the efficiency and performance of model training.

3.3 Limitations of computing resources

As model complexity increases and data scales expand, traditional computing resources often fail to meet the demands of training and inference. This results in excessively long training times, high energy consumption, and even the inability to complete large-scale model training. Particularly in deep learning, complex neural network architectures require substantial computational power, which places extremely high demands on hardware resources.

To address these challenges, computational mathematics provides multiple effective solutions. Distributed computing stands as a crucial strategy, significantly boosting efficiency through parallel processing of computational tasks across multiple nodes. Frameworks like MapReduce and Spark are widely used in big data processing and model training. These frameworks optimize resource utilization by sharding data for parallel processing, thereby reducing training time. Furthermore, parameter server architectures enhance distributed systems by centrally managing model parameters and coordinating gradient updates across computing nodes. The efficiency and stability of training. GPU acceleration technology serves as another crucial solution. Graphics Processing Units (GPUs), with their powerful parallel computing capabilities, are particularly suited for intensive computational tasks like matrix operations. In deep learning, GPU acceleration can significantly reduce model training time and enhance efficiency. Model compression and quantization techniques also provide effective solutions for addressing computational resource limitations. Through methods like pruning and low-rank decomposition, these techniques reduce the number of parameters and computational complexity, thereby decreasing storage and computational demands. Pruning technology removes redundant computations by eliminating insignificant weights in the model, to improve the reasoning speed of the model. Quantization technology can significantly reduce the storage space and calculation amount of the model while maintaining the prediction accuracy of the model by converting the weight and activation value of the model from high precision to low precision.

4. Conclusion

In summary, computational mathematics plays a vital role in optimizing artificial intelligence algorithms. It provides a solid theoretical foundation and diverse optimization methods for AI systems, driving continuous technological advancement through ongoing applications and developments. However, computational mathematics in algorithm optimization also faces challenges that require further exploration. Looking ahead, with the development of emerging technologies and the expanding application scenarios of AI, computational mathematics will see broader and deeper integration into AI algorithm optimization. This integration will bring both new opportunities and challenges to the evolution of artificial intelligence technology.

References

[1] Liu B. Application of Computational Mathematics in Artificial Intelligence Algorithm Optimization. Microcomputer. 2024;(06):106-8.

[2] Zhang J, Wei HT. Application Research of Artificial Intelligence in University Laboratory Management. Comput Knowl Technol. 2025;21(11):51-3.

[3] Li Y, Wu JC, Lin X. AI Empowering Teaching Innovation in Environmental Design Courses at Universities. In: China Architectural Society Interior Design Branch, editor. 2024 Interior Design Proceedings. Shihezi University College of Literature and Arts; Changchun University of Humanities; Northeast Normal University School of Fine Arts; 2024. p. 183-7.

[4] Huang LW, Zhang JP, Dong HY, et al. Research on Teaching Reform of Advanced Mathematics Courses in Applied Universities Based on Artificial Intelligence. J Mudanjiang Educ Inst. 2024;(08):72-6.

[5] Xu ZM, Meng RX. Practice of AI Innovative Talent Cultivation in Western Universities' Mathematics Departments. Educ Teach Forum. 2024;(22):173-6.

[6] Xia MX, Wu CY, Lin AQ. Research Review and Prospects of Artificial Intelligence Empowering Ideological and Political Education in Universities—Based on CiteSpace Knowledge Graph Data Analysis. J Luohe Vocat Tech Coll. 2023;22(03):97-101.

[7] Chen LJ, Pan ZJ, Chen XR. AI-Based Performance Prediction Model for University Students. Comput Syst Appl. 2023;32(06):212-20.

[8] Yang B, Liang S. A Brief Discussion on AI Talent Cultivation in Local Universities. J Guiyang Univ (Nat Sci Ed). 2022;17(01):99-103.

[9] Zhong MY, Hong Y, Su JM, et al. Exploring Educational Programs for Local Universities in the AI Era. Sci Technol Rev. 2020;(26):7-8.

[10] Hu W, Mei XE, Yang W. Research on Interactive Teaching Methods for AI-Based Algorithm Courses. Comput Knowl Technol. 2025;21(16):157-9.