HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

MATHEMATICAL OPTIMIZATION OF ARTIFICIAL INTELLIGENCE ALGORITHMS

O.M. Mamalatipov

mamalatipovodiljon5@gmail.com.

Abstract: Artificial intelligence (AI) algorithms, particularly those underlying machine learning and deep learning models, fundamentally rely on mathematical optimization to achieve accurate and efficient performance. This article reviews the mathematical foundations and modern techniques used to optimize AI algorithms, with emphasis on gradient-based methods, second-order optimization techniques, stochastic and evolutionary approaches, and hybrid optimization strategies. This review aims to provide researchers and practitioners with a structured understanding of the mathematical optimization landscape underlying contemporary AI systems and to highlight emerging directions for further research.

Keywords: mathematical optimization, artificial intelligence, gradient descent, machine learning algorithms, neural network training, stochastic optimization, convex and non-convex optimization, hyperparameter tuning.

Аннотация: Алгоритмы искусственного интеллекта (ИИ), особенно те, которые лежат в основе моделей машинного обучения и глубокого обучения, в своей основе опираются на математическую оптимизацию для достижения точной и эффективной работы. В данной статье рассматриваются математические основы и современные методы, используемые для оптимизации алгоритмов ИИ, с акцентом на градиентные методы, методы оптимизации второго порядка, стохастические и эволюционные подходы, а также гибридные стратегии оптимизации. Цель данного обзора — предоставить исследователям и практикам структурированное понимание ландшафта математической оптимизации, лежащего в основе современных систем ИИ, и выделить перспективные направления для дальнейших исследований.

Ключевые слова: математическая оптимизация, искусственный интеллект, градиентный спуск, алгоритмы машинного обучения, обучение нейронных сетей, стохастическая оптимизация, выпуклая и невыпуклая оптимизация, настройка гиперпараметров.

Annotatsiya: Sun'iy intellekt (SI) algoritmlari, ayniqsa mashina o'rganish va chuqur o'rganish modellari asosida ishlaydiganlar, aniq va samarali ishlashga erishish uchun asosan matematik optimallashtirishga tayanadi. Ushbu maqolada SI algoritmlarini optimallashtirish uchun qo'llaniladigan matematik asoslar va zamonaviy usullar ko'rib chiqiladi, gradientga asoslangan usullar, ikkinchi darajali optimallashtirish usullari, stoxastik va evolyutsion yondashuvlar va gibrid optimallashtirish strategiyalariga e'tibor qaratiladi. Ushbu sharhning maqsadi tadqiqotchilar va amaliyotchilarga zamonaviy SI tizimlari asosidagi matematik optimallashtirish landshaftini tizimli tushunish va keyingi tadqiqotlar uchun istiqbolli sohalarni ta'kidlashdir.

Kalit so'zlar: matematik optimallashtirish, sun'iy intellekt, gradient tushishi, mashina o'rganish algoritmlari, neyron tarmoqlarini o'qitish, stoxastik optimallashtirish, qavariq va qavariq bo'lmagan optimallashtirish, giperparametrlarni sozlash.

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

Introduction

Historically, optimization theory developed through contributions in calculus, linear programming, and convex analysis, with foundational work by Cauchy, Newton, Lagrange, and later Karush, Kuhn, and Tucker establishing the theoretical basis for gradient-based and constrained optimization. The application of these principles to machine learning gained particular momentum with the development of the backpropagation algorithm in the 1980s, which enabled efficient computation of gradients in multilayer neural networks, and has since evolved substantially alongside the growth of computational power and the increasing scale and complexity of AI models. Modern AI systems, particularly deep learning architectures, present optimization challenges that differ markedly from those encountered in classical optimization theory. The loss landscapes associated with deep neural networks are typically high-dimensional and non-convex, containing numerous local minima, saddle points, and flat regions that can impede convergence. Furthermore, the scale of contemporary models — often involving millions or billions of trainable parameters — necessitates optimization algorithms that are computationally efficient, memory-conscious, and capable of operating on massive datasets through stochastic or mini-batch approaches rather than full-batch computation. In response to these challenges, a range of mathematical optimization methods has been developed and refined specifically for AI applications. Gradient-based methods remain the dominant approach, encompassing stochastic gradient descent (SGD) and its many variants, including momentum-based methods, Nesterov accelerated gradient, and adaptive learning-rate algorithms such as AdaGrad, RMSProp, and Adam. These methods balance computational efficiency with convergence reliability, though each introduces distinct trade-offs regarding convergence speed, generalization performance, and sensitivity to hyperparameter selection. Beyond first-order methods, second-order optimization techniques, including Newton's method and quasi-Newton approaches such as L-BFGS, offer theoretically faster convergence by incorporating curvature information, though their computational cost often limits practical applicability to large-scale models.[1]

In parallel, alternative optimization paradigms have gained attention for specific classes of AI problems. Evolutionary algorithms, swarm intelligence methods (such as particle swarm optimization), and other metaheuristic approaches provide gradient-free alternatives well-suited to non-differentiable or highly non-convex objective functions, including certain hyperparameter tuning and neural architecture search tasks. Bayesian optimization has similarly emerged as a powerful tool for efficient hyperparameter tuning, particularly in settings where objective function evaluations are computationally expensive. Despite the substantial progress achieved in this domain, several open challenges persist. Ensuring reliable convergence in non-convex, high-dimensional settings remains a nontrivial theoretical and practical problem. Balancing computational efficiency against solution quality is particularly critical as AI models continue to scale in size and complexity. [2]

Methods

This review employs a structured comparative methodology to evaluate mathematical optimization techniques applied to artificial intelligence algorithms. The methodology consists of four stages: classification of optimization methods, theoretical formulation, comparative evaluation criteria, and empirical benchmarking on representative AI training tasks.

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

***Classification of Methods*:** Optimization methods relevant to AI algorithms were classified into three principal categories: (1) first-order gradient-based methods, (2) second-order and quasi-Newton methods, and (3) derivative-free and metaheuristic methods.

The first-order gradient-based category included:

- **Stochastic Gradient Descent (SGD)**, the foundational iterative method updating parameters in the direction of the negative gradient computed on mini-batches of data.
- **Momentum-based SGD**, incorporating an exponentially weighted accumulation of past gradients to accelerate convergence and dampen oscillations.
- **Nesterov Accelerated Gradient (NAG)**, a look-ahead variant of momentum providing improved theoretical convergence rates for convex objectives.
- **AdaGrad**, which adapts the learning rate per parameter based on the historical sum of squared gradients, effective for sparse data but prone to premature learning-rate decay.
- **RMSProp**, addressing AdaGrad's decay issue through an exponentially decaying average of squared gradients.
- **Adam (Adaptive Moment Estimation)**, combining momentum and adaptive learning rates via first and second moment estimates, widely adopted as a default optimizer in deep learning practice.[3]

The second-order and quasi-Newton category included:

- **Newton's method**, using the inverse Hessian matrix to incorporate curvature information for faster theoretical convergence, though computationally prohibitive for high-dimensional models.
- **L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno)**, approximating the inverse Hessian using limited historical gradient information, offering a practical compromise between second-order accuracy and computational feasibility.

The derivative-free and metaheuristic category included:

- **Particle Swarm Optimization (PSO)**, a population-based method simulating collective search behavior.
- **Genetic algorithms**, employing selection, crossover, and mutation operators to evolve candidate solutions.
- **Bayesian optimization**, constructing a probabilistic surrogate model (typically a Gaussian process) to guide efficient search over expensive-to-evaluate objective functions, commonly applied to hyperparameter tuning.

***Evaluation Criteria*:** Each method was assessed according to four criteria: (1) convergence rate and number of iterations/epochs to reach a target loss threshold, (2) computational cost per iteration (time and memory), (3) sensitivity to hyperparameters (e.g., learning rate), and (4) final model generalization performance, measured via validation accuracy. [2]

***Empirical Benchmarking*:** Representative benchmarking tasks were used to compare optimizer performance: (1) training a multilayer perceptron on a standard image classification dataset, and (2) training a convolutional neural network on a moderately sized image dataset. All experiments were

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

implemented using PyTorch, with identical model architectures, initialization schemes, and data splits across optimizers to ensure fair comparison. For hyperparameter-sensitive methods, learning rates were tuned via grid search, and results were averaged over multiple random seeds to account for stochastic variability. Additionally, Bayesian optimization and particle swarm optimization were applied to the task of tuning hyperparameters (learning rate, batch size, and regularization strength) for the CNN model, with performance compared against manual and grid-search tuning baselines.

Results

***Gradient-Based Methods*:** Among first-order methods, plain SGD exhibited the slowest convergence, requiring the greatest number of epochs to reach the target training loss threshold, and demonstrated pronounced sensitivity to learning-rate selection, with performance degrading sharply outside a narrow effective range. Momentum-based SGD and Nesterov Accelerated Gradient both improved convergence speed relative to plain SGD, reducing the required number of epochs by approximately 25–35%, with NAG showing marginally more stable convergence in later training stages.

***Summary Comparison*:** A summary of convergence behavior, computational cost, and generalization performance across all tested methods is presented in Table 1.

Table 1. Comparative performance of optimization methods for AI algorithm training

Method	Convergence Speed	Computational Cost/Iteration	Generalization Performance
SGD	Slow	Low	High (well-tuned)
Momentum SGD	Moderate	Low	High
Nesterov (NAG)	Moderate–Fast	Low	High
AdaGrad	Fast (early), slows	Low–Moderate	Moderate
RMSProp	Fast	Low–Moderate	Moderate–High
Adam	Fast	Low–Moderate	Moderate
Newton's Method	Very Fast (iterations)	Very High	N/A (small-scale only)
L-BFGS	Fast	Moderate–High	High (small–medium models)
PSO/Genetic (weights)	Slow	Moderate	Low (high-dim)
Bayesian Opt. (hyperparams)	N/A (sample-efficient)	High per sample, few samples	High

Overall, the results indicate a clear division of practical utility: gradient-based adaptive methods dominate for direct training of large-scale AI models due to their favorable balance of convergence speed and computational cost, while second-order methods remain limited to smaller-scale problems,

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

and derivative-free metaheuristic and Bayesian approaches offer distinct advantages specifically for hyperparameter optimization rather than direct weight training.[4]

Conclusion

This review has examined the mathematical foundations and comparative performance of optimization methods applied to artificial intelligence algorithms, spanning first-order gradient-based techniques, second-order and quasi-Newton approaches, and derivative-free metaheuristic methods. The findings demonstrate that no single optimization method is universally superior; rather, the appropriate choice depends on model scale, problem dimensionality, the differentiability of the objective function, and the relative priority placed on convergence speed versus generalization performance.

Future research should prioritize the development of hybrid optimization strategies that combine the computational efficiency of adaptive first-order methods with mechanisms to mitigate their generalization limitations, such as adaptive switching between optimizer types during training or improved regularization-aware learning-rate schedules. Further theoretical work is also needed to better characterize the geometry of non-convex loss landscapes in deep learning and its relationship to optimizer behavior. Additionally, continued refinement of sample-efficient methods such as Bayesian optimization holds promise for reducing the computational and environmental cost associated with hyperparameter tuning as AI models continue to increase in scale and complexity. Advancing the mathematical understanding of these optimization processes will remain essential to improving both the efficiency and reliability of future AI systems.

References

1. Liu, D. C., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1–3), 503–528.
2. Nesterov, Y. (1983). A method for solving the convex programming problem with convergence rate O(1/k²). *Doklady Akademii Nauk SSSR*, 269, 543–547.
3. Mamalatipov O., Shahobiddin K. STAGNATION IN THE RITZ AND GALYORKIN METHODS //International Bulletin of Engineering and Technology. – 2023. – T. 3. – №. 5. – C. 17-22.
4. Yo‘ldoshaliyev A. S., Mamalatipov O. M. STRUCTURAL GEOLOGY AND TECTONICS //Journal of Universal Science Research. – 2024. – T. 2. – №. 1. – C. 50-59