Proceedings of the 4th International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/2026.BA33212

A Survey of Optimization Methods in Machine Learning: From Gradient Descent to Convex Optimization

Shizheng Song

School of Mathematics & Statistics, The University of New South Wales, Sydney, Australia
18505461622@163.com

Abstract. Optimization plays a fundamental role in machine learning, as most learning tasks can be formulated as the minimization of a loss function. From classical gradient descent to modern convex optimization theory, optimization algorithms have continuously evolved to meet the demands of large-scale data and high-dimensional models. This paper reviews the development of optimization methods in machine learning, focusing on gradient descent and its variants, stochastic optimization, and convex optimization theory. Through literature analysis, this study examines the theoretical foundations, convergence properties, and practical applications of these methods. Specifically, the research addresses three key questions: how gradient-based methods have evolved, what advantages convex optimization provides, and what challenges arise in non-convex optimization. The paper concludes that convex optimization offers strong theoretical guarantees, while gradient-based algorithms dominate practical large-scale machine learning tasks, especially in deep learning.

Keywords: Machine learning, gradient descent, stochastic optimization, convex optimization, non-convex optimization

1. Introduction

Optimization problems constitute one of the central foundations of machine learning. Most machine learning models can be formulated as minimizing an objective function that measures the discrepancy between predicted outputs and observed data [1]. This formulation transforms learning into a mathematical optimization process in which model parameters are iteratively updated to reduce prediction errors. Early machine learning research relied primarily on classical gradient-based optimization methods, particularly batch gradient descent [2], which computes parameter updates using the entire dataset. However, with the rapid growth of data scale and model complexity, especially in deep learning, optimization methods have undergone significant evolution [2,3]. Stochastic optimization techniques, accelerated gradient methods, and various strategies for handling non-convex problems have been developed to address both computational and theoretical challenges. At the same time, convex optimization theory provides a rigorous mathematical framework for analyzing learning algorithms. Convex optimization problems possess desirable properties, such as the guarantee that any local optimum is also a global optimum and well-defined convergence behavior [4]. Many traditional models, including linear regression and support vector machines, can be formulated as convex optimization problems [5]. Nevertheless, modern deep

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.BA33212

neural networks typically involve highly non-convex objective functions, leading to challenges such as local minima, saddle points, and unstable convergence behavior [6,7].

This paper reviews optimization methods used in machine learning from both theoretical and practical perspectives. It examines the evolution of gradient-based methods, analyzes the theoretical advantages of convex optimization, and discusses the challenges associated with non-convex optimization in modern machine learning systems. Through this analysis, this paper aims to provide a clearer understanding of the development and significance of optimization algorithms in modern machine learning.

2. Optimization framework in machine learning

2.1. Optimization modeling in machine learning

Machine learning problems are typically formulated under the empirical risk minimization (ERM) framework [1,8]. Given a dataset $(x_i, y_i), i = 1,2,\ldots,n,$ the objective is to minimize the empirical risk defined as

$$\frac{1}{n}\sum_{i=1}^{n} l\left(f\left(x_i\right),\theta\right) + \lambda R\left(\theta\right) \tag{1}$$

where l denotes the loss function, and $R(\theta)$ represents a regularization term controlling model complexity [8]. ERM aims to minimize the average loss over the training dataset, which measures the discrepancy between predicted outputs and true labels. Loss functions vary depending on the learning task. For example, mean squared error is commonly used in regression problems, while cross-entropy loss is widely applied in classification tasks. Regularization methods play an essential role in improving model generalization by preventing overfitting. Common techniques include L1 and L2 regularization. L1 regularization encourages sparsity in model parameters, whereas L2 regularization penalizes large parameter magnitudes and promotes smoother parameter distributions. These methods transform the optimization problem into a penalized minimization problem [9].

2.2. Mathematical characteristics of optimization problems

The mathematical properties of optimization problems strongly affect algorithm design. Consider a standard optimization problem:

$$\min_{\theta\in\mathbb{R}^d} f\left(\theta\right) \tag{2}$$

The gradient represents the direction of steepest ascent of the objective function, which is defined as

$$\nabla f\left(\theta\right) = \left(\frac{\partial f}{\partial \theta_1}, \frac{\partial f}{\partial \theta_2}, \ldots, \frac{\partial f}{\partial \theta_d}\right)^T \tag{3}$$

Optimization algorithms typically update parameters along the negative gradient direction to minimize objective functions. The Hessian matrix captures second-order curvature information and is defined as

DOI: 10.54254/2755-2721/2026.BA33212

$$\nabla^2 f(\theta) = \begin{bmatrix} \frac{\partial^2 f}{\partial \theta_1^2} & \cdots & \frac{\partial^2 f}{\partial \theta_1 \partial \theta_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial \theta_d \partial \theta_1} & \cdots & \frac{\partial^2 f}{\partial \theta_d^2} \end{bmatrix} \tag{4}$$

First-order methods rely only on gradient information, whereas second-order methods use curvature information to achieve faster convergence, though at higher computational cost. Convexity plays a fundamental role in optimization theory. A function is convex if any line segment between two points on its graph lies above the function. Convex functions guarantee that any local minimum is also a global minimum, which greatly simplifies optimization.

The distinction between first-order and second-order methods also determines computational complexity. First-order methods require relatively low computational cost per iteration, while second-order methods require computation and inversion of the Hessian matrix, which becomes expensive in high-dimensional settings.

3. Gradient-based optimization algorithms

3.1. Gradient descent

Gradient descent is one of the most fundamental optimization algorithms used in machine learning [2]. It computes the gradient of the objective function using the entire dataset at each iteration and updates model parameters accordingly. The standard update rule is given by

$$\theta_{k+1} = \theta_k - \eta \nabla f(\theta_k) \tag{5}$$

where $\eta > 0$ denotes the step size (learning rate). In machine learning applications, gradient descent is typically implemented as batch gradient descent, where gradients are computed using the entire training dataset. This approach provides stable convergence and guarantees optimal solutions for convex problems.

However, batch gradient descent becomes computationally inefficient for large datasets because each iteration requires processing all training samples. The convergence behavior of gradient descent also depends on the choice of the learning rate and the condition number of the objective function. Improper learning rate selection may lead to slow convergence or divergence.

3.2. Stochastic gradient descent and mini-batch methods

Stochastic Gradient Descent (SGD) was proposed to address the computational limitations of gradient descent [3]. Instead of using the entire dataset, SGD updates parameters using a single training sample at each iteration. This approach significantly reduces computational cost and enables large-scale learning.

Although SGD introduces noise into gradient estimation, it often converges faster in practice and can escape shallow local minima [10]. However, its convergence trajectory is less stable compared to traditional gradient descent.

In practice, mini-batch gradient descent represents a compromise between batch and stochastic methods. It computes gradients using small subsets of data, balancing computational efficiency and

DOI: 10.54254/2755-2721/2026.BA33212

convergence stability. Mini-batch methods are widely used in deep learning due to their parallel computation capability.

3.3. Accelerated methods

Standard gradient descent often suffers from slow convergence in ill-conditioned optimization problems [11]. Accelerated methods improve convergence speed by incorporating momentum terms that accumulate past gradient information. The momentum method introduces a velocity variable that smooths parameter updates:

$$v_{t+1} = \gamma v_t + \eta \nabla f(\theta_t),\ \theta_{t+1} = \theta_t - v_{t+1} \tag{6}$$

Nesterov accelerated gradient further improves convergence by evaluating gradients at predicted future positions [12]. These methods achieve faster convergence rates and improve optimization performance in high-dimensional parameter spaces.

3.4. Adaptive learning rate algorithms

Traditional gradient-based methods such as batch gradient descent and stochastic gradient descent typically rely on a manually specified global learning rate [13]. Selecting an appropriate learning rate is often challenging in practice. A large learning rate may cause divergence, whereas a small learning rate can lead to slow convergence. Moreover, in high-dimensional parameter spaces, different parameters may exhibit different sensitivities to updates, making a single global learning rate inefficient. To address these limitations, adaptive learning rate algorithms have been developed. These methods automatically adjust step sizes for individual parameters based on historical gradient information.

AdaGrad is an early representative approach that scales updates using accumulated gradient information [14]. It assigns larger effective learning rates to parameters associated with infrequent features, making it particularly suitable for sparse data. However, a common limitation is that the accumulated gradient term continually increases, causing the effective learning rate to shrink over time which may slow progress in later training stages. Adam is a widely used adaptive optimizer that combines adaptive scaling with momentum-like behavior [14]. In practice, Adam often accelerates optimization and stabilizes training, particularly in large-scale deep learning. Overall, adaptive methods are mainly valued for convenience and robustness in optimization, while careful learning-rate scheduling with SGD remains a strong baseline in many applications.

4. Convex and non-convex optimization: theory and challenges

4.1. Theoretical significance of convex optimization

Convex optimization provides strong theoretical guarantees for machine learning algorithms. For convex objective functions, any local minimum is also a global minimum, which eliminates the ambiguity often present in non-convex problems. This fundamental property allows gradient-based algorithms, when properly configured, to converge toward a unique optimal solution. As a result, the optimization process becomes more predictable, and convergence behavior can be rigorously analyzed. In particular, for smooth convex functions, first-order methods enjoy well-established convergence rates, while strong convexity further guarantees linear convergence under appropriate step-size conditions.

DOI: 10.54254/2755-2721/2026.BA33212

Convex optimization theory also provides powerful analytical tools such as duality theory and the Karush–Kuhn–Tucker (KKT) conditions for constrained optimization problems. Duality theory allows complex primal problems to be analyzed through their dual formulations, which often provide deeper structural insights and computational advantages. The KKT conditions provide necessary optimality conditions, and under certain regularity assumptions, sufficient conditions for optimality in convex optimization problems. These theoretical principles form the mathematical backbone of many machine learning optimization procedures [4].

Many classical machine learning models correspond to convex optimization problems, including linear regression, logistic regression, and support vector machines. In these models, the objective functions are convex with respect to the model parameters, enabling global optimal solutions to be efficiently obtained. This convex structure contributes to strong theoretical interpretability, stable training behavior, and reliable generalization performance. Efficient optimization algorithms such as gradient descent, coordinate descent, and interior-point methods can therefore be applied with well-understood convergence guarantees [5]. Consequently, convex optimization remains a cornerstone of traditional machine learning methodologies and continues to influence modern algorithm design.

4.2. Challenges in non-convex optimization

Modern machine learning models, particularly deep neural networks, typically involve highly non-convex objective functions. Non-convex optimization landscapes contain multiple critical points, including local minima, saddle points, and flat regions, which complicate optimization.

Local minima correspond to points where the gradient vanishes but the solution is not globally optimal. Early research suggested that local minima posed the primary challenge in training neural networks. However, more recent studies indicate that high-dimensional optimization landscapes contain numerous saddle points, which represent stationary points with both positive and negative curvature. Formally, a saddle point satisfies $\nabla f(\theta) = 0, and\ \nabla^2 f(\theta)$ has both positive and negative eigenvalues.

While the Hessian matrix has both positive and negative eigenvalues. Saddle points are particularly problematic because gradient-based methods may converge slowly near these regions. The gradient magnitude becomes small, leading to slow progress even though the solution is not optimal. In high-dimensional spaces, saddle points are more prevalent than poor local minima, making them a major obstacle to efficient optimization. Non-convex optimization also suffers from issues such as gradient vanishing and gradient explosion, particularly in deep neural networks. These problems hinder parameter updates and affect training stability.

4.3. Practical strategies in deep learning

To address non-convex optimization challenges, several practical techniques have been developed to improve training stability and convergence efficiency in deep neural networks. Parameter initialization methods influence the starting point of optimization and significantly affect convergence behavior. Poor initialization may lead to slow convergence or cause the optimization process to become trapped near undesirable critical points. Modern initialization schemes, such as Xavier initialization and He initialization, aim to maintain the variance of activations and gradients across layers, thereby mitigating gradient vanishing and explosion problems and improving training efficiency.

Learning rate scheduling is another widely used strategy that adjusts the step size during training. Instead of using a constant learning rate, scheduling methods gradually reduce the step size over

DOI: 10.54254/2755-2721/2026.BA33212

iterations to stabilize convergence near optimal solutions. Common approaches include step decay, exponential decay, and cosine annealing strategies. These methods enable large updates during early training stages to accelerate learning while ensuring fine-grained adjustments in later stages.

Batch normalization has also been introduced to improve optimization performance in deep neural networks. By normalizing intermediate layer activations, batch normalization reduces internal covariate shift and smooths the optimization landscape. This normalization improves gradient flow, accelerates convergence, and allows the use of higher learning rates. Furthermore, normalization techniques often enhance generalization performance by introducing a regularization effect.

In addition to these techniques, gradient clipping is commonly used to prevent excessively large parameter updates caused by exploding gradients, particularly in recurrent neural networks. Proper network architecture design and regularization methods, such as dropout, also contribute to stabilizing optimization by reducing overfitting and improving the structure of the loss landscape.

Recent theoretical research has explored conditions under which gradient-based methods can escape saddle points and converge to approximate local minima. These studies suggest that stochastic noise in gradient estimation helps optimization algorithms avoid strict saddle points in high-dimensional spaces. These findings provide theoretical support for the empirical success of gradient-based methods in deep learning despite the inherent non-convexity of the objective functions.

5. Conclusion

Optimization methods form the foundation of machine learning algorithms and have evolved significantly. This paper reviewed gradient-based optimization methods, examined the theoretical role of convex optimization, and analyzed challenges associated with non-convex optimization.

Gradient-based methods provide practical scalability and computational efficiency, making them suitable for large-scale machine learning tasks. Convex optimization offers strong theoretical guarantees, including global optimality and convergence properties. However, modern machine learning models typically involve non-convex objectives, which introduce challenges such as saddle points and unstable convergence.

A trade-off exists between theoretical guarantees and practical performance. While convex optimization provides rigorous analysis, gradient-based methods remain dominant in practice due to their computational efficiency. Future research should focus on improving optimization efficiency, developing algorithms that better handle non-convex landscapes, and strengthening theoretical understanding of large-scale optimization.

Optimization continues to be a central research area in machine learning, and advances in optimization theory will play a crucial role in improving model performance and expanding applications.

References

[1] Liu, X., Qi, H., Jia, S., et al. (2025) Recent advances in optimization methods for machine learning: a systematic review. Mathematics, 13(13): 2210.

[2] Sun, R. (2020) Optimization for deep learning: Theory and algorithms. IEEE Transactions on Signal Processing, 68: 4893–4906.

[3] Wang, M., Fu, W., He, X., et al. (2020) A survey on large-scale machine learning. IEEE Transactions on Knowledge and Data Engineering, 34(6): 2574-2594.

[4] Cui, Y., Paquette, C., & Scheinberg, K. (2020) Multi-composite nonconvex optimization framework for training deep neural networks. SIAM Journal on Optimization, 30(1): 596–629.

DOI: 10.54254/2755-2721/2026.BA33212

[5] Ding, K., Li, J., Toh, K. C. (2025) Nonconvex stochastic Bregman proximal gradient method with application to deep learning. Journal of Machine Learning Research, 26(39): 1-44.
[6] Allen-Zhu, Z., Li, Y., Song, Z. (2019) A convergence theory for deep learning via over-parameterization. International conference on machine learning. PMLR, 242-252.
[7] Elnady, S. M., El-Beltagy, M., Radwan, A. G., et al. (2025) A comprehensive survey of fractional gradient descent methods and their convergence analysis. Chaos, Solitons & Fractals, 194: 116154.
[8] Bian, K., Priyadarshi, R. (2024) Machine learning optimization techniques: a survey, classification, challenges, and future research issues. Archives of Computational Methods in Engineering, 31(7): 4209-4233.
[9] Foret, P., Kleiner, A., Mobahi, H., et al. (2020) Sharpness-aware minimization for efficiently improving generalization. arXiv preprint arXiv: 2010.01412.
[10] Gower, R. M., Schmidt, M., Bach, F., et al. (2020) Variance-reduced methods for machine learning [J]. Proceedings of the IEEE, 108(11): 1968-1983.
[11] Yang, L., Shami, A. (2020) On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing, 415: 295-316.
[12] Zhang, W., Cao, W., Ji, J. (2024) Deep Learning-Enabled Non-convex Optimization: Encoder-Decoder Forecasting and ADMM Solver. Proceedings of the International Conference on Computer Vision and Deep Learning, 1-6.
[13] Wang, X., Yan, L., Zhang, Q. Research on the application of gradient descent algorithm in machine learning. 2021 international conference on computer network, electronic and automation (ICCNEA). IEEE, 2021: 11-15.
[14] Das, R., Acharya, A., Hashemi, A., et al. (2022) Faster non-convex federated learning via global and local momentum. Uncertainty in Artificial Intelligence. PMLR, 496-506.
[15] Fernández, J. G., Ahmad, N., van Gerven, M. (2025) A Unified Perspective on Optimization in Machine Learning and Neuroscience: From Gradient Descent to Neural Adaptation. arXiv preprint arXiv: 2510.18812.
[16] Abdulkadirov, R., Lyakhov, P., Nagornov, N. (2023) Survey of optimization algorithms in modern neural networks. Mathematics, 11(11): 2466.
[17] Lascu, R. A., Majka, M. B. (2025) Non-convex entropic mean-field optimization via Best Response flow. arXiv preprint arXiv: 2505.22760.
[18] Zhang, Y., Yang, Q. (2021) A survey on multi-task learning. IEEE transactions on knowledge and data engineering, 34(12): 5586-5609.