

Perspective

Evolution and Role of Optimizers in Training Deep Learning Models

By XiaoHao Wen , Member, IEEE, and MengChu Zhou , Fellow, IEEE

To perform well, deep learning (DL) models have to be trained well. Which optimizer should be adopted? We answer this question by discussing how optimizers have evolved from traditional methods like gradient descent to more advanced techniques to address challenges posed by high-dimensional and non-convex problem space. Ongoing challenges include their hyperparameter sensitivity, balancing between convergence and generalization performance, and improving interpretability of optimization processes. Researchers continue to seek robust, efficient, and universally applicable optimizers to advance the field of DL across various domains.

A. Introduction to DL and Optimization

The rapid advancement of DL has significantly promoted various applications, from computer vision and natural language processing to speech recognition and beyond [1]–[4]. At the heart of this progress lies the development of sophisticated DL architectures and models. These powerful models have achieved remarkable success in learning complex patterns and representations from vast amounts of data, enabling breakthroughs in a wide range of applications. However, their success heavily relies on effective training methods, i.e., optimizers. Optimization meets significant challenges due to the high-dimensional and non-convex nature of the problem space [5]. Traditional optimizers, such as gradient descent, often struggle with slow convergence and the propensity to get trapped in local minima [6]. We see a strong need for increasingly sophisticated and powerful optimizers.

The journey of optimizer evolution began with the introduction of stochastic gradient descent (SGD) [7] as shown in Fig. 1. It has brought stochasticity into an optimization process, enabling faster convergence and better generalization. However, the uniform learning rate across all parameters in SGD limits its adaptability to the diverse learning dynamics of different model components. This limitation motivates the development of adaptive learning rate methods, such as Adagrad [8], RMSprop [9], and Adam [10]. These optimizers

This work was partially supported by the Guangxi Universities and Colleges Young and Middle-aged Teachers' Scientific Research Basic Ability Enhancement Project (2023KY0055). Corresponding author: MengChu Zhou.

Citation: X. H. Wen and M. C. Zhou, "Evolution and role of optimizers in training deep learning models," *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 10, pp. 2039–2042, Oct. 2024.

X. H. Wen is with Guangxi Normal University, Guilin 541004, China (e-mail: wenxiao@gxnu.edu.cn).

M. C. Zhou is with the School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China, and also with the Helen and John C. Hartmann Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JAS.2024.124806

introduce parameter-specific learning rates, allowing for more efficient traversal of an optimization landscape.

B. Overview of DL Models

The diverse landscape of DL models form the backbone of today's artificial intelligence (AI) applications. Each model is designed to excel in specific domains or to address particular challenges in data processing and representation learning. These models have evolved significantly over time with various architectures emerging to tackle different types of problems. Convolutional neural networks (CNNs) have revolutionized computer vision tasks, from image classification to object detection and segmentation. Their representatives include LeNet, AlexNet, VGGNet, and ResNet, each contributing to the advancement of the field.

Recurrent neural networks (RNNs) and their variants, such as long short-term memory (LSTM) and gated recurrent unit (GRU), have proven effective in processing sequential data, making them particularly useful for tasks involving time series or natural language. The introduction of the Transformer [11] has led to significant advancements in natural language processing. Transformer-based models like BERT, GPT, and T5 have achieved outstanding results in various natural language processing (NLP) tasks.

In addition to these mainstream architectures, several specialized and emerging models have gained attention. Graph neural networks (GNNs) are designed to process data represented as graphs, making them suitable for tasks involving relational data, social networks, or molecular structures. Tensor-based Neural Networks have shown promise in handling complex, multi-dimensional data structures through tensor distribution regression based on 3D conventional neural networks. Dendritic neuron model (DNM) represents a biologically inspired approach to neural network design. Unlike traditional artificial neurons that perform a simple weighted sum of inputs, DNMs incorporate more complex processing within each neuron, mimicking the dendritic computations observed in biological neurons. DNMs offer an interesting alternative to traditional DL architectures, potentially leading to more efficient and biologically plausible AI systems.

The effectiveness of these diverse architectures heavily depends on the optimizers employed during training. An optimizer can significantly impact not only the training speed but also the generalization performance of the trained model. For instance, the implicit regularization effect of SGD may be particularly beneficial for CNNs, while adaptive methods like Adam for large Transformer models.

C. Evolution of Optimizers

Evolution From SGD to Adaptive Methods: The optimization landscape in DL has witnessed significant transformations since the introduction of SGD. Yet its uniform learning rate across all parameters limited its adaptability to diverse learning dynamics. This limitation leads to the development of adaptive learning rate methods. They are



Fig. 1. Evolution of optimizers for deep learning model training.

well represented by Adagrad, RMSprop, and Adam. These optimizers employ parameter-specific learning rates, resulting in more efficient traversal of the optimization landscape than SGD. Adam, in particular, has gained widespread popularity due to its ability to adapt the learning rate for each parameter based on first and second moment estimates of the gradient.

Generalization Challenge: Despite the significant improvements brought by adaptive optimizers, they often exhibit a trade-off between fast initial convergence and the ability to generalize well to unseen data. This observation has spurred research into hybrid optimization strategies, e.g., SWATS [12] and Adabound [13], which aim to combine the benefits of adaptive methods with the stability and generalization capabilities of SGD.

Regularization-Based Optimizers: Recent advancements have focused on incorporating regularization techniques directly into an optimization process. Adaptive learning-rate with momentum (AdamW) [14] has gained popularity for its ability to combine the benefits of Adam with improved generalization through decoupled weight decay.

Sharpness-Aware Minimization: A significant breakthrough in optimizer design comes with the introduction of sharpness-aware minimization (SAM) [15] and its adaptive variant called ASAM [16]. These optimizers explicitly seek flat minima in the loss landscape, leading to improved generalization. SAM has shown remarkable performance across various tasks, particularly in improving model robustness and out-of-distribution generalization.

Emerging Optimizers: Several novel optimizers have been proposed to address the limitations of existing methods and push the boundaries of optimization in DL. These emerging optimizers introduce innovative techniques such as combining adaptive methods with normalized directions (Adan [17]), using predicted changes in loss for adaptive learning rates (AdaBelief [18]), and leveraging second-order information for faster convergence and better generalization (Sophia [19]). These advancements aim to improve performance, stability, and efficiency across a wide range of DL tasks, particularly in large-scale models.

D. Heuristic Optimization Algorithms

Heuristic optimization algorithms have gained significant attention in recent years due to their robust exploration capabilities and ability to avoid local optima. They offer several advantages over gradient-based methods, such as global search capabilities, parallelization potential, flexibility in handling non-differentiable or discontinuous objective functions, and efficient hyperparameter optimization. These features make them particularly useful for complex, non-convex optimization problems in DL.

Heuristic algorithms have been successfully applied to various aspects of DL, including training models, optimizing hyperparameters, and designing model architectures. Population-based heuristic algorithms, such as Particle Swarm Optimization and Evolutionary Algorithms, have been leveraged for their inherent parallelizability, enabling efficient optimization of large-scale models [20]–[23]. Furthermore, heuristic algorithms have been employed to optimize the architecture and parameters of DNNs. By utilizing heuristic optimization techniques, researchers have been able to effectively train and optimize DNN-based models for various machine learning tasks

SAM: Sharpness-aware minimization
ASAM: Adaptive sharpness-aware minimization
Adan: Adaptive Nesterov momentum
Lion: Least-squares Interpolation optimizer with noise
Sophia: Second-order clipped stochastic

[24], [25]. This approach has shown promising results in improving the performance of DNNs.

Despite the promising outlook, heuristic algorithms face several challenges when applied to DL. The computational cost associated with function evaluations can be substantial, especially for large-scale models. Scalability issues may arise as the problem dimensionality increases, and the theoretical foundations of heuristic algorithms are less robust compared to gradient-based methods. Addressing these challenges through improved algorithm design, scalability enhancements, and deeper theoretical analysis will be crucial for the continued success of heuristic algorithms in DL optimization.

E. Hyperparameter Sensitivity

Hyperparameter tuning has long been a problem. It often impacts research progress and the reliability and reproducibility of experimental results, emerging as a major bottleneck in the performance and widespread application of DL models.

1) Learning Rate Selection: The selection of an appropriate learning rate is one of the most challenges. A well-chosen rate can accelerate model convergence, while a poor choice may lead to training failure. Even with adaptive methods such as Adam or AdamW, finding the optimal one and initial value often necessitates extensive grid searches or complex learning rate scheduling strategies. This process is typically time-consuming, labor-intensive, and computationally expensive, resulting in significant energy waste and environmental burden.

More problematically, a learning rate strategy that performs well on one task may fail in other tasks or models. This forces us to repeat this tedious parameter tuning process when facing different tasks. For instance, a learning rate setting that excels in computer vision tasks may completely fail in natural language processing tasks, presenting substantial challenges due to this cross-domain non-transferability.

2) Parameters in Adam-Like Optimizers: Parameters β_1 and β_2 in Adam and its variants (e.g., AdamW and RAdam) can greatly impact model performance. They control the exponential decay rates of the first and second moment estimates, respectively, directly affecting the optimizer's efficiency in utilizing gradients. However, their optimal values vary depending on the task and model architecture, making it difficult to find a universally applicable setting.

More complexly, there exists a subtle interplay between them. For example, β_1 's change affects the optimal β_2 value, and vice versa. This intricate interdependence makes the tuning process complex and unpredictable. In practice, we often adopt default values ($\beta_1 = 0.9$ and $\beta_2 = 0.999$), potentially missing their best settings. How to quickly find such best settings for different tasks remains open.

3) Weight Decay and Adaptive Learning Rates: Weight decay, as an important regularization technique, often has an overlooked interaction effect with adaptive learning rates. This interaction can lead to unexpected optimization behaviors. For example, in Adam, the implementation method of weight decay significantly affects the model's convergence performance. Traditional L2 regularization in Adam may lead to suboptimal results, which has given rise to variants such as AdamW.

This complex interaction requires us to not only adjust learning rates and momentum parameters but also consider the impact of weight decay, greatly increasing the complexity of parameter tuning.

More challengingly, different layers of weights may require different decay rates, introducing the concept of layer-adaptive weight decay and further complicating the optimization process.

4) Reproducibility Problem: The sensitivity of hyperparameters affects the reproducibility of DL outcomes. Researchers working on the same problem may arrive at drastically different results due to subtle differences in hyperparameter tuning. For instance, in some cases, merely changing the random seed can lead to significant differences in model performance. This instability makes it exceptionally difficult to compare the true performance of different methods. More seriously, this irreproducibility may lead to erroneous research conclusions, misleading subsequent research directions.

To mitigate this problem, some researchers advocate for standardized hyperparameter search processes or automated hyperparameter optimization methods. However, these approaches often require substantial computational resources, which may be challenging for the projects with limited resources. How to ensure experimental reproducibility with limited resources has become an urgent problem to be solved.

F. Generalization Performance

The issue of generalization performance in optimizers has long been a focal point. It impacts the practical application of DL models. This problem has not only garnered widespread attention in academia but has also become a critical consideration for industry professionals when deploying DL models in real-world scenarios.

1) Adaptive Methods vs. SGD: While adaptive methods enjoy rapid convergence, numerous studies have demonstrated that their generalization performance often falls short of carefully-tuned SGD with momentum. This phenomenon has been consistently observed across various research efforts. For instance, in computer vision tasks, ResNet models trained using SGD typically outperform those trained with Adam on test sets. This presents a challenging dilemma: should they prioritize faster training speeds or superior model performance? As rapid training is highly valued, this question becomes particularly pressing. Especially for resource-constrained projects, striking a balance between these two objectives poses a significant challenge.

2) Implicit Regularization Effect of SGD: SGD is believed to possess a form of implicit regularization effects, which may account for its superior generalization performance over adaptive methods. However, the precise mechanisms underlying this remain elusive, hampering our ability to design optimizers that can achieve both rapid convergence and excellent generalization performance. Understanding and replicating this characteristic of SGD has emerged as a key challenge in optimizer research. Currently, researchers are delving into this issue from both theoretical and practical perspectives, a pursuit that promises to be highly valuable.

3) Performance and speed Trade-offs With SAM: SAM and ASAM methods have shown promising results in enhancing model generalization performance. However, the substantial computational overhead of SAM could potentially become a severe bottleneck in large-scale model training. This is particularly problematic when dealing with extensive datasets or training ultra-large models, as the additional computational burden may significantly contribute to increased training time and costs. Exploring ways to simplify SAM's computational process or identifying alternative methods that achieve similar effects with greater computational efficiency is a crucially important topic.

4) Balancing Task Specificity and Universality: Different tasks may require distinct optimization strategies to achieve optimal generalization performance. For example, optimizer configurations that excel in natural language processing tasks may prove less effective in computer vision ones. Nevertheless, in practical applications, there is often a desire for a "universal" optimizer capable of performing well across various tasks. This demand is particularly pronounced in industrial settings, where maintaining multiple optimizer versions for different tasks would increase system complexity and maintenance costs. Finding a balance between task specificity and universality

remains challenging.

G. Interpretability Problem

As DL technology rapidly advances, the complexity of optimizers has increased significantly, rendering their behavior increasingly opaque. This lack of transparency is evident not only in traditional gradient-based optimizers but also in emerging heuristic ones. This phenomenon, known as the "black box problem" of optimizers, poses several challenges in practical applications:

1) Inscrutable Decision-Making Processes: Modern adaptive and heuristic algorithms often show potential in addressing various tasks. However, their internal decision-making processes remain largely enigmatic. This lack of interpretability hinders our ability to accurately diagnose and effectively address training issues. For instance, when model convergence is unusually slow or overfitting occurs, it becomes challenging to determine whether the issue lies with the optimizer or model architecture. In the case of heuristic algorithms, while they effectively explore complex parameter space, we often fail to explain why certain search paths yield superior results. This opacity is particularly problematic in fields demanding high levels of safety and interpretability, such as medical diagnostics and financial risk management, potentially impeding DL's deployment.

2) Complex Interactions Between Optimizer Behavior and Model Representations: The interplay between optimizer behavior and the representations learned by DL models is intricate and poorly understood. This issue becomes even more complex when using heuristic algorithms, as their search strategies may lead to entirely different parameter trajectories. Consequently, it becomes difficult to ascertain whether performance improvements stem from enhanced optimization processes or superior feature representations. In practical applications, this problem is particularly pronounced as identical model architectures may learn different feature representations when using different optimizers. For heuristic algorithms, due to their stochastic nature and global search characteristics, even runs with identical initial conditions may yield significantly different results. Such phenomena make it challenging for us to determine whether to focus on improving the optimizer, algorithm design, or data preprocessing to enhance model performance.

3) Impediments to Transfer Learning and Meta-Learning: The lack of interpretability in optimizer behavior also presents challenges for the application of transfer learning and meta-learning. Establishing principled methods to transfer optimization strategies learned from one task to another proves difficult. This issue is even more pronounced when using heuristic algorithms because their search strategies are often tailored to specific problems, making them difficult to transfer directly to other tasks. Furthermore, the lack of deep understanding of the algorithms' internal mechanisms hampers our ability to design meta-learning algorithms that can automatically adapt to different tasks.

4) Balancing Computational Cost and Interpretability: While heuristic algorithms can sometimes find superior solutions, they typically require more function evaluations, translating to higher computational cost. As we strive for better performance, we face the challenge of balancing computational efficiency with model interpretability. A fine trade-off is particularly needed when large-scale DL models are trained.

H. Conclusion and Future Perspectives

The evolution of optimizers is on-going with a mission to cope with great challenges posed by high-dimensional, non-convex optimization landscapes. From the foundational SGD to cutting-edge approaches like Sharpness-Aware Minimization and Sophia, optimizers have continuously adapted to meet the demands of DL models. This journey has been marked by significant innovations, including adaptive learning rate methods, regularization-based techniques, and the exploration of heuristic algorithms. However, persistent challenges remain, particularly in areas such as hyperparameter sensitivity, generalization performance, and interpretability. Making desired trade-off between fast convergence and excellent generalization, han-

dling complex interactions between optimizer behavior and model representations, and balancing well computational efficiency and interpretability motivate researchers to advance the field of optimizers. Looking ahead, the key areas for future research include:

- 1) Developing self-adaptive optimizers with improved generalization capabilities.
- 2) Enhancing the theoretical understanding of optimizer behavior in DL contexts.
- 3) Exploring task-specific optimization strategies while maintaining cross-domain applicability.
- 4) Improving the scalability and efficiency of advanced optimization techniques for large-scale models.
- 5) Enhancing the interpretability of optimization processes to facilitate model diagnostics and reliability.

As DL continues to evolve, addressing these challenges and exploring new optimization paradigms will be crucial in unlocking its full potential across various domains and applications. Our next work is to compare extensively the optimizers in terms of their performance in training some specific deep learning models, for example, powerful dendritic neuron models [24], [26]–[28]. We hope to gain sufficient insights 1) to guide engineers in selecting right optimizers when they face their particular applications; and 2) to motivate researchers to invent robust, parameter-free and universally effective optimizers to train various DL models across many application domains, especially autonomous driving [29], [30], Internet of Behaviors [31], and Industry 5.0/Automation 5.0 [32], [33].

REFERENCES

- [1] Z. Zhang *et al.*, “Mapping network-coordinated stacked gated recurrent units for turbulence prediction,” *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 6, pp. 1331–1341, 2024.
- [2] H. Liu, *et al.*, “Aspect-based sentiment analysis: A survey of deep learning methods,” *IEEE Trans. Computational Social Systems*, vol. 7, no. 6, pp. 1358–1375, Dec. 2020.
- [3] H. Wu *et al.*, “A PID-incorporated latent factorization of tensors approach to dynamically weighted directed network analysis,” *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 3, pp. 533–546, 2022.
- [4] W. Xu *et al.*, “Transformer-based macroscopic regulation for highspeed railway timetable rescheduling,” *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 9, pp. 1822–1833, 2023.
- [5] I. Goodfellow *et al.*, *Deep Learning*. MIT press, 2016.
- [6] S. Ruder, “An overview of gradient descent optimization algorithms,” arXiv preprint arXiv: 1609.04747, 2016.
- [7] H. Robbins *et al.*, “A stochastic approximation method,” *The Annals of Mathematical Statistics*, pp. 400–407, 1951.
- [8] J. Duchi *et al.*, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, p. 7, 2011.
- [9] T. Tieleman, “Lecture 6.5-RMSprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA: Neural Networks for Machine Learning*, vol. 4, p. 2, 2012.
- [10] D. P. Kingma *et al.*, “Adam: A method for stochastic optimization,” arXiv preprint arXiv: 1412.6980, 2014.
- [11] A. Vaswani *et al.*, “Attention is all you need,” *Advances in Neural Information Processing Systems*, p. 30, 2017.
- [12] N. S. Keskar *et al.*, “Improving generalization performance by switching from Adam to SGD,” arXiv preprint arXiv: 1712.07628, 2017.
- [13] L. Luo *et al.*, “Adaptive gradient methods with dynamic bound of learning rate,” in *Proc. Int. Conf. Learning Representations*, 2019.
- [14] I. Loshchilov *et al.*, “Decoupled weight decay regularization,” arXiv preprint arXiv: 1711.05101, 2017.
- [15] P. Foret *et al.*, “Sharpness-aware minimization for efficiently improving generalization,” arXiv preprint arXiv: 2010.01412, 2020.
- [16] J. Kwon *et al.*, “ASAM: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks,” in *Proc. Int. Conf. Machine Learning*, 2021, pp. 5905–5914.
- [17] X. Xie *et al.*, “Adan: Adaptive nesterov momentum algorithm for faster optimizing deep models,” arXiv preprint arXiv: 2208.06677, 2022.
- [18] J. Zhuang *et al.*, “Adabelief optimizer: Adapting stepsizes by the belief in observed gradients,” in *Proc. Conf. Neural Information Processing Systems*, 2020.
- [19] H. Liu *et al.*, “Sophia: A scalable stochastic second-order optimizer for language model pre-training,” arXiv preprint arXiv: 2305.14342, 2023.
- [20] J. Chen *et al.*, “Hierarchical particle swarm optimization-incorporated latent factor analysis for large-scale incomplete matrices,” *IEEE Trans. Big Data*, vol. 8, no. 6, pp. 1524–1536, 2022.
- [21] M. Cui *et al.*, “Surrogate-assisted autoencoder-embedded evolutionary optimization algorithm to solve high-dimensional expensive problems,” *IEEE Trans. Evolutionary Computation*, vol. 26, no. 4, pp. 676–689, 2022.
- [22] G. Wei *et al.*, “A hybrid probabilistic multiobjective evolutionary algorithm for commercial recommendation systems,” *IEEE Trans. Computational Social Systems*, vol. 8, no. 3, pp. 589–598, 2021.
- [23] J. Bi *et al.*, “Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization,” *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3774–3785, 2021.
- [24] S. Gao *et al.*, “Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction,” *IEEE Trans. Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 601–614, 2018.
- [25] Y. Yu *et al.*, “Improving dendritic neuron model with dynamic scalefree network-based differential evolution,” *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 1, pp. 99–110, 2021.
- [26] X. Luo *et al.*, “Interpretability diversity for decision-tree-initialized dendritic neuron model ensemble,” *IEEE Trans. Neural Networks and Learning Systems*, doi: 10.1109/TNNLS.2023.3290203, 2023.
- [27] Y. Yu *et al.*, “Improving dendritic neuron model with dynamic scale-free network-based differential evolution,” *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 1, pp. 99–110, Jan. 2022.
- [28] S. Gao *et al.*, “Fully complex-valued dendritic neuron model,” *IEEE Trans. Neural Networks and Learning Systems*, vol. 34, no. 4, pp. 2105–2118, Apr. 2023.
- [29] F.-Y. Wang, “Intelligent vehicles from your HomePorts to underwaters and low altitude airspaces: SLAM for smart societies,” *IEEE Trans. Intelligent Vehicles*, vol. 9, no. 2, pp. 3092–3105, Feb. , 2024.
- [30] G. Yuan *et al.*, “An autonomous vehicle group cooperation model in an urban scene,” *IEEE Trans. Intelligent Transportation Systems*, vol. 24, no. 12, pp. 13852–13862, Dec. , 2023.
- [31] Q. Zhao *et al.*, “A tutorial on Internet of behaviors: Concept, architecture, technology, applications, and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1227–1260, Secondquarter , 2023.
- [32] S. Lou *et al.*, “Human-cyber-physical system for Industry 5.0: A review from a human-centric perspective,” *IEEE Trans. Automation Science and Engineering*, doi: 10.1109/TASE.2024.3360476, 2024.
- [33] L. Vlasic *et al.*, “Automation 5.0: The key to systems intelligence and Industry 5.0,” *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 8, pp. 1723–1727, Aug. , 2024.

ABOUT THE AUTHOR

XiaoHao Wen Bio of XiaoHao Wen can be found at <https://ieeexplore.ieee.org/author/37089229997>.

MengChu Zhou Bio of MengChu Zhou can be found at <https://ieeexplore.ieee.org/author/37273591600>.