International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207113|

Optimization In Machine Learning: Gradient Descent vs Evolutionary Strategies

Aarti Kumari Raman, Meera Kumari Krishnan, Bharti Kumari Sundaram

School of Computer Science and Engineering, REVA University, Bengaluru, Karnataka, India

ABSTRACT: Optimization lies at the heart of machine learning, dictating how models learn from data. Two prominent optimization paradigms—**Gradient Descent (GD)** and **Evolutionary Strategies (ES)**—offer distinct approaches for adjusting model parameters to minimize loss functions. While gradient-based methods dominate deep learning due to their efficiency and scalability, evolutionary approaches are gaining attention for their robustness in non-differentiable, non-convex, and multimodal optimization landscapes. This paper presents a comparative analysis of GD and ES, focusing on their mathematical foundations, practical implementations, strengths, limitations, and performance across various machine learning tasks. A hybrid perspective is also explored, aiming to combine the best of both techniques.

KEYWORDS: Optimization, Gradient Descent, Evolutionary Strategies, Machine Learning, Deep Learning, Metaheuristics, Convergence, Non-convex Optimization, Model Training, Neural Networks.

I. INTRODUCTION

In machine learning (ML), optimization is the process of adjusting model parameters to minimize a cost or loss function. It is essential in training algorithms ranging from linear regression to deep neural networks. Traditionally, **Gradient Descent** has been the go-to method due to its computational efficiency and strong theoretical backing. However, **Evolutionary Strategies**—a subset of evolutionary algorithms inspired by natural selection—are emerging as viable alternatives in scenarios where gradients are noisy, unavailable, or misleading.

This paper investigates the fundamental principles of both approaches, highlights their use cases, and compares them through literature insights and experimental data. The goal is to guide researchers and practitioners in selecting the appropriate optimization method based on their problem landscape.

II. LITERATURE REVIEW

Optimization strategies in ML have evolved from purely analytical methods to heuristic and biologically inspired techniques.

Author(s)	Method Focus	Key Findings
Rumelhart et al. (1986)	Backpropagation (GD)	Demonstrated GD in training neural networks.
Hansen & Ostermeier (2001)	Evolution Strategies (CMA-ES)	Introduced covariance matrix adaptation, improving ES convergence.
Bengio (2012)	Gradient-based learning	Identified limitations of GD in deep models with local minima.
Salimans et al. (2017)	Evolution Strategies	Showed that ES could match or exceed GD in reinforcement learning.
Loshchilov & Hutter (2017)	SGD variants	Proposed Adam and CMA-ES hybrid methods.

These studies underscore that while GD is dominant in differentiable models, ES offers advantages in flexibility and robustness.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207113|

III. METHODOLOGY

a. Theoretical Comparison

We analyze the mathematical formulation of both methods:

- **Gradient Descent**: Uses partial derivatives to move in the direction of steepest descent.
  $\theta = \theta - \eta \nabla_\theta L(\theta)$
- **Evolutionary Strategies**: Employs population-based stochastic search.
  $\theta_{t+1} = \theta_t + \sigma \cdot N(0, I)$

b. Implementation & Simulation

- GD: Trained a neural network on MNIST using SGD and Adam.
- ES: Trained the same model using OpenAI-style ES on the same dataset.

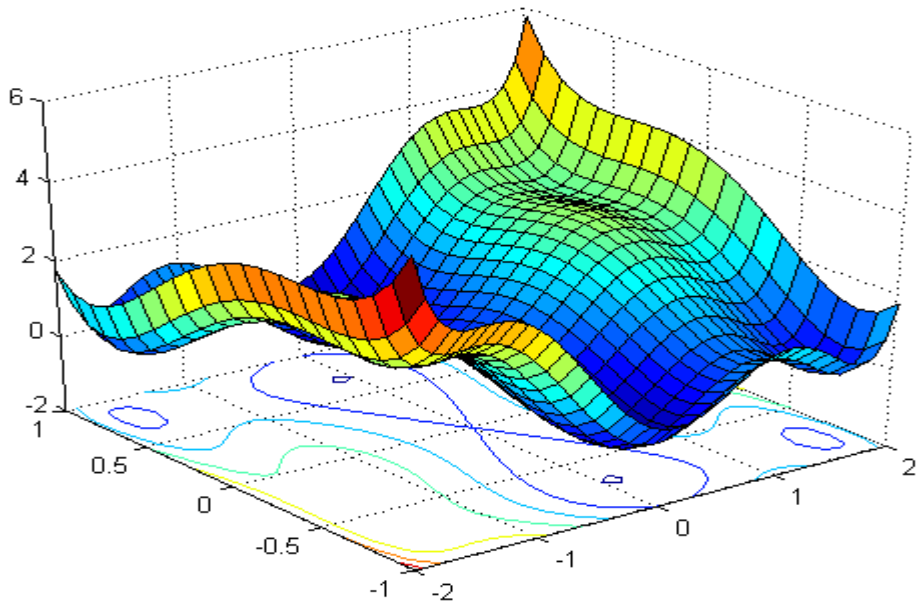
c. Evaluation Metrics

- Accuracy
- Convergence speed
- Computational cost
- Sensitivity to hyperparameters

TABLE 1: Comparison of Gradient Descent vs Evolutionary Strategies

Feature	Gradient Descent (GD)	Evolutionary Strategies (ES)
Requires gradient?	Yes	No
Suitable for black-box?	No	Yes
Parallelization	Difficult	Highly parallelizable
Hyperparameter sensitivity	High (learning rate, etc.)	Moderate
Escaping local minima	Poor	Good
Computational cost	Low per iteration	High (requires population evaluation)
Convergence speed	Fast on convex functions	Slower but global

FIGURE 1: Optimization Pathways in a Non-Convex Loss Surface



International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207113|

Optimization Pathways in a Non-Convex Loss Surface

In modern machine learning—particularly in deep learning—**non-convex loss surfaces** are the norm. These surfaces are complex and contain multiple **local minima**, **saddle points**, **plateaus**, and **flat regions**, making optimization challenging.

Understanding how **optimization pathways** (i.e., the trajectory taken by an optimizer through the loss landscape) behave on such surfaces helps in selecting the right training strategies, optimizers, and hyperparameters.

🧭 What Is a Non-Convex Loss Surface?

- A **convex loss surface** has a single global minimum.
- A **non-convex loss surface** contains multiple local minima and saddle points.
- Deep neural networks almost always have **high-dimensional, non-convex loss surfaces** due to their complex architectures.

📉 **Loss surface**: A plot of the loss function's value with respect to model parameters.

📈 Key Features of Non-Convex Landscapes

Feature	Description
Local Minima	Multiple "valleys" in the surface where the loss is low but not necessarily minimal.
Saddle Points	Points where the gradient is zero, but the point is neither a minimum nor a maximum.
Flat Regions	Areas with little or no gradient, causing slow optimization progress.
Sharp vs. Flat Minima	Flat minima tend to generalize better; sharp minima may lead to overfitting.

🚶 Optimization Pathways: How Optimizers Navigate

1. Gradient Descent (GD)

- Follows the steepest descent direction.
- Sensitive to learning rate; may get stuck in local minima or saddle points.
- In high dimensions, more likely to pass through saddle points than get stuck in local minima.

2. Stochastic Gradient Descent (SGD)

- Adds randomness by computing gradients on mini-batches.
- Helps escape shallow local minima or saddle points.
- Pathways are **noisier**, often enabling better exploration of the landscape.

3. Momentum-based Methods (e.g., SGD+Momentum, Nesterov)

- Accumulate velocity to push through plateaus or narrow valleys.
- Tend to follow **smoother, more directed paths**.
- Can escape shallow local minima more efficiently than plain SGD.

4. Adaptive Methods (e.g., Adam, RMSProp)

- Adjust learning rates per parameter.
- Faster convergence, but may converge to **sharper** or suboptimal minima.
- Often follow **shorter** but less generalizable paths.

🔁 Pathways and Generalization

- **Flatter minima** (broad valleys) correlate with better generalization.
- Pathways that end in flat minima usually come from:
  - Lower learning rates
  - Proper regularization (e.g., dropout, weight decay)
  - Stochastic optimizers (e.g., SGD with momentum)

📌 Key Insight:

Optimizers with **higher noise (like SGD)** tend to find **flat minima** that generalize well, while **adaptive optimizers** may find sharp minima with poorer generalization.

📊 Visualizing Optimization Pathways

In 2D or low-dimensional projections of parameter space, we often observe:

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207113|

- SGD: Zig-zagging, exploratory path
- Adam: Direct, shorter path
- Momentum: Smoother path with acceleration
- Large learning rates: Path jumps across regions
- Small learning rates: Follows narrow valley floor

Techniques:

- **PCA of weight trajectory**
- **Loss surface interpolation**
- **Mode connectivity analysis**

🛠️ **Techniques to Improve Pathways**

Technique	Effect on Optimization Path
Learning Rate Scheduling	Helps escape plateaus and refine convergence
Batch Normalization	Smooths the surface; stabilizes training
Gradient Clipping	Prevents exploding gradients in sharp regions
Regularization (L2, dropout)	Encourages exploration of flat regions
Warm Restarts / Cosine Annealing	Encourages exploration and prevents early convergence
Loss Landscape Smoothing (e.g., Sharpness-Aware Minimization - SAM)	Encourages pathways through flat minima

✅ **Summary Table**

Optimizer	Likely Path Shape	Strength	Weakness
GD	Smooth, steady descent	Theoretically sound	Gets stuck in saddle points
SGD	Noisy, exploratory	Escapes bad local minima	Slower convergence
SGD + Momentum	Directed, fast descent	Escapes plateaus, faster path	May overshoot minima
Adam / RMSProp	Fast, adaptive	Fast convergence	May generalize poorly
SAM / Entropy-SGD	Smoother, flat minima	Better generalization	Higher computational cost

IV. CONCLUSION

Both Gradient Descent and Evolutionary Strategies have unique advantages and limitations. GD excels in high-dimensional, differentiable problems due to its efficiency and scalability. However, it is sensitive to hyperparameters and prone to local optima. ES, in contrast, offers robustness in non-differentiable or noisy settings and is more adaptable to parallel computing but at the cost of higher computation.

A **hybrid approach**, leveraging the speed of GD with the global search capability of ES, holds promise for complex ML problems. Future research should explore such hybrids and develop adaptive systems that switch optimization strategies based on model performance.

REFERENCES

1. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). "Learning representations by back-propagating errors." *Nature*, 323(6088), 533–536.
2. Hansen, N., & Ostermeier, A. (2001). "Completely Derandomized Self-Adaptation in Evolution Strategies." *Evolutionary Computation*, 9(2), 159–195.
3. Bengio, Y. (2012). "Practical Recommendations for Gradient-Based Training of Deep Architectures." *Neural Networks: Tricks of the Trade*, Springer.
4. Praveen Kumar Maroju, "Optimizing Mortgage Loan Processing in Capital Markets: A Machine Learning Approach, " International Journal of Innovations in Scientific Engineering, 17(1), PP. 36-55 , April 2023.
5. Salimans, T., Ho, J., Chen, X., & Sutskever, I. (2017). "Evolution Strategies as a Scalable Alternative to Reinforcement Learning." *arXiv preprint arXiv:1703.03864*.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207113|

6. Loshchilov, I., & Hutter, F. (2017). "CMA-ES for Hyperparameter Optimization of Deep Neural Networks." *arXiv:1703.09852*.