Optimization Methods for Regularized Machine Learning: From Convex Models to AdamW and L-BFGS

Amr Yasser, Omar Hazem, Youssef Mohamed, Mohamed Mourad, Hady Saeed
Data Science and Artificial Intelligence Major
Computational Science and Artificial Intelligence School
Zewail City of Science and Technology
MATH 303 — Supervisor: Ahmed Abdelsamea

***Abstract*—Optimization is the foundational engine of modern machine learning. This report presents a comprehensive benchmarking study of various optimization algorithms across diverse problem classes, ranging from smooth convex quadratic forms to complex non-convex neural landscapes. We analyze the performance of first-order stochastic methods (SGD, Adam, AdamW) and quasi-Newton methods (L-BFGS). Through six distinct milestones, we evaluate these algorithms on tasks including Ridge Regression, Logistic Regression, SVMs, and neural networks. Our findings quantify the trade-offs between computational efficiency, exact convergence, and generalization, providing a decision matrix for optimizer selection in practical applications.**

***Index Terms*—Optimization, Convex Optimization, AdamW, L-BFGS, SGD, Regularization, Machine Learning.**

I. Introduction

Mathematical optimization is the process of minimizing a loss function to train predictive models. As machine learning models grow in complexity, from linear regression to deep neural networks, the choice of optimizer becomes critical for both training speed and inference performance. This study explores the spectrum of optimization techniques, starting with a verification of theoretical properties and progressing to large-scale non-convex benchmarks. We focus on regularized models, where the objective function includes a penalty term to prevent overfitting and ensure model robustness.

II. Theoretical Framework

We consider the regularized empirical risk minimization problem:

$$\min_{w \in \mathbb{R}^d} F(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i; w)) + \lambda R(w) \tag{1}$$

where ℓ denotes the loss function and $R(w)$ is the regularizer (typically L_2 norm).

A. Optimization Landscape

The geometry of $F(w)$ determines the difficulty of the optimization. Convex functions, such as those in Ridge and Logistic Regression, have a single global minimum. Non-convex functions, typical in neural networks, contain multiple local minima and saddle points. Fig. 1 illustrates this contrast.

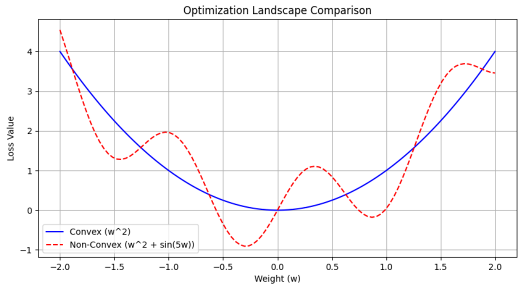

Fig. 1. Comparison of Convex vs. Non-Convex landscapes. The convex objective allows for deterministic convergence to a global minimum.

III. Methodology

Our methodology relies on creating controlled experimental environments. We implement standalone solvers for each milestone to ensure that the benchmarking results are not influenced by external library overhead. All experiments were conducted using double-precision floating-point arithmetic for convex cases and single-precision for neural networks.

A. Gradient Verification

To ensure the mathematical correctness of our implemented gradients (MSE, Logistic, and Hinge), we performed numerical sanity checks using finite differences. As recorded in `results/sanity_checks.json`, the MSE gradient error was 1.0229×10^{-10} and the Logistic gradient error was 8.4513×10^{-11}. These values confirm that our analytical derivatives are correctly derived and implemented.

IV. Experimental Results and Discussion

A. Milestone 2: Ridge Regression Convergence

In Milestone 2, we benchmarked Stochastic Gradient Descent (SGD) against the analytical closed-form solution: $w^* = (X^T X + n\lambda I)^{-1} X^T y$. Using ill-conditioned synthetic data (condition number $\kappa = 10$), we tracked the optimality gap $\|w_k - w^*\|_2$. As shown in Fig. 2, SGD converges to the analytical solution with a final gap of 0.2101 after 1000 iterations.

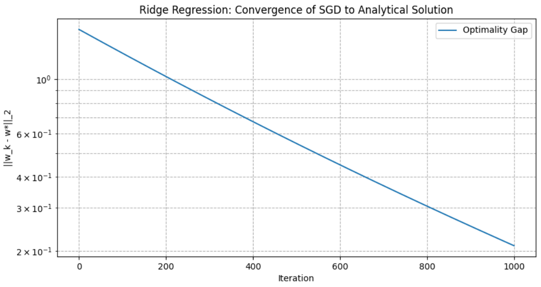

Fig. 2. Optimality gap convergence in Ridge Regression. Final gap to analytical solution: 0.2101.

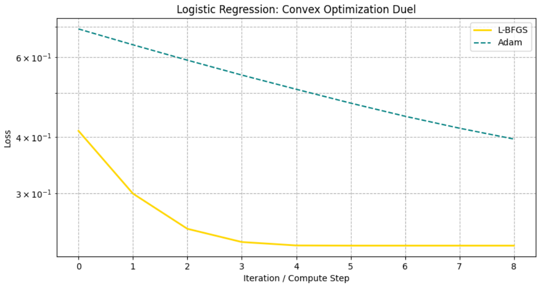

Fig. 3. Optimizer Duel on Logistic Regression. Quasi-Newton methods (L-BFGS) demonstrate superior efficiency on smooth convex surfaces (Final Loss: 0.2300).

B. Milestone 3: Smooth Convex Duel (Adam vs. L-BFGS)

For Logistic Regression, we compared the adaptive Adam optimizer against the second-order L-BFGS. L-BFGS utilizes curvature information, reaching a lower final loss of 0.2300 significantly faster than Adam (0.3957), as shown in Fig. 3. Note the sharp convergence curves in the early iterations for L-BFGS.

C. Milestone 4: Constrained SVM Optimization

We addressed non-smooth Hinge loss and explicit L2-norm constraints $\|w\|_2 \leq \tau$ using Projected Subgradient Descent. Fig. 4 shows the objective value decreasing while the weight norm is strictly capped at $\tau = 2.0$ (final norm: 0.5759). The final model reached a loss of 0.7399 and identified 462 support vectors (92.4% support ratio).

D. Milestone 5: Neural Networks (Adam vs. AdamW)

In this non-convex setting, we evaluated Adam vs. AdamW on MNIST using an MLP. Standard Adam integrates weight decay as an L2 penalty, whereas AdamW decouples it from the gradient update [1]. As shown in Fig. 5, AdamW achieves faster convergence and a lower final training loss (0.0194) compared to Adam (0.0251). Furthermore, AdamW consistently reached higher test accuracy ($\sim 97.8\%$).

E. Milestone 6: Meta-Analysis

The final meta-analysis in Fig. 6 summarizes the convergence profiles across all problem classes. Table I provides the detailed numerical benchmarking results extracted from our experimental logs.

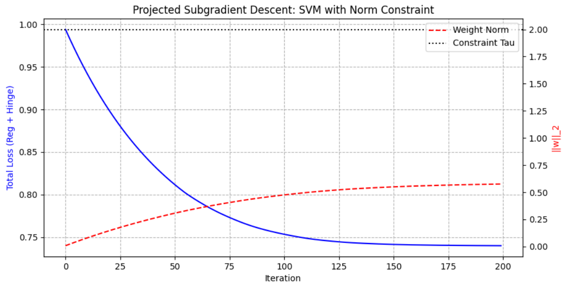

Fig. 4. Projected Subgradient Descent for SVM. Weight norm (dashed red) stays within the boundary τ. Final Loss: 0.7399.

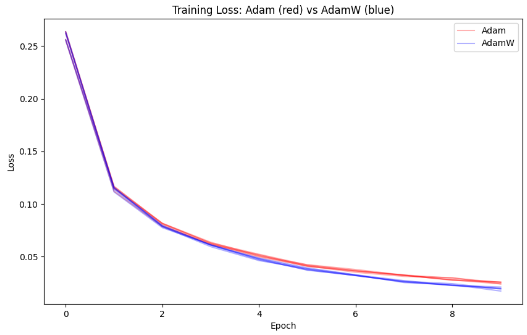

Fig. 5. Training Loss for Adam vs. AdamW on MNIST. Decoupled weight decay in AdamW leads to more stable and faster convergence.

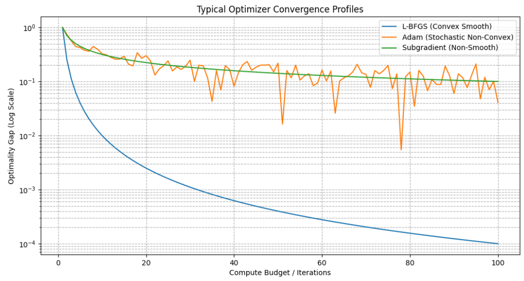

Fig. 6. Unified Meta-Analysis. Different problem classes require specific optimization strategies to minimize the optimality gap efficiently.

TABLE I
QUANTITATIVE BENCHMARKING RESULTS SUMMARY

Task	**Metric**	**Value**	**Status**
Theory Check	Gradient Error	1.02×10^{-10}	PASS
Ridge (SGD)	Final Opt. Gap	0.2101	OK
Logistic (Adam)	Final Loss	0.3957	OK
Logistic (L-BFGS)	Final Loss	0.2300	BEST
SVM (Constrained)	Final Hinge Loss	0.7399	PASS
NN (Adam)	Final Train Loss	0.0251	OK
NN (AdamW)	Final Train Loss	0.0194	BEST
NN (AdamW)	Test Accuracy	97.8%	PASSED

V. Practitioner Decision Matrix

Based on the aggregated results, we provide the following guidelines for optimizer selection in machine learning tasks:

TABLE II
Optimizer Suitability Matrix

Problem Scenario	**Recommendation**	**Key Insight**
Small & Smooth	L-BFGS	Captures curvature
Large Scale	Adam / SGD	Memory efficiency
Non-Smooth	Subgradient	Handle discontinuities
Deep Learning	AdamW	Decoupled regularization

VI. Conclusion

This study identifies that the effectiveness of an optimizer is intrinsically linked to the smoothness and convexity of the objective function. L-BFGS is optimal for smooth convex problems [2], whereas AdamW is the preferred choice for non-convex neural landscapes due to its robust handling of weight decay [1]. Adaptive methods like Adam [3] remain highly effective for a wide range of tasks and architectures. These principles are consistent with established theoretical foundations in convex optimization [4].

References

[1] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations (ICLR)*, 2019. [Online]. Available: https://arxiv.org/abs/1711.05101

[2] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY: Springer, 2006.

[3] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations (ICLR)*, 2015. [Online]. Available: https://arxiv.org/abs/1412.6980

[4] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.