From Regularized (Convex) Regression to AdamW, L-BFGS, and Constrained Training: A Review and Benchmarking Study

Amr Yasser (202301043) — Omar Hazem (202300800) — Yousef Mohamed (202300220)
Hady Saeed (202301707) — Mohamed Mourad (202302348)

Zewail City of Science and Technology, Egypt
MATH 303 — Linear and Nonlinear Programming
Instructor: Prof. Ahmed Abdelsamea

***Abstract*—Optimization is the computational core of modern machine learning, yet many commonly used training algorithms in deep learning are often presented without clear connections to the classical linear/nonlinear programming (LP/NLP) concepts that explain their behavior. This project proposes a review-and-benchmark study that unifies popular learning objectives and optimizers under a single optimization viewpoint: empirical risk minimization with regularization and (optional) constraints. We connect convex models—ridge regression, logistic regression, and linear SVMs—to LP/NLP tools including convexity, duality, Karush–Kuhn–Tucker (KKT) conditions, conditioning, and quasi-Newton approximations. We then extend the comparison to a small nonconvex neural network to examine how first-order, adaptive, and quasi-Newton methods behave when classical assumptions break down. Beyond a narrative survey, the project's contribution is a reproducible experimental protocol: consistent stopping criteria, compute budgets, and robustness checks across optimizers (SGD/Momentum, Adam/AdamW, and L-BFGS), with controlled sensitivity analysis over learning rates, regularization strength, and random seeds. The expected outcome is a course-aligned report that doubles as a practitioner-oriented review with clear "if-then" guidance for choosing optimization methods in AI workflows.**

***Index Terms*—Convex optimization, empirical risk minimization, regularization, KKT conditions, quasi-Newton methods, L-BFGS, AdamW, constrained training, benchmarking.**

I. Introduction (Background & Brief Literature Review)

Many supervised learning pipelines can be written as optimization problems: minimize an empirical loss (data fit) plus regularization (model complexity control). In convex settings, LP/NLP theory provides strong guarantees (global optimality), interpretable optimality certificates (duality/KKT), and well-studied numerical behavior under ill-conditioning and finite precision [1], [2]. In contrast, modern deep learning typically solves nonconvex objectives using stochastic first-order methods (SGD and momentum variants) and adaptive methods in the Adam family, which are often effective but sensitive to hyperparameters and may behave differently in terms of generalization [3]–[6].

Several foundational works connect machine learning practice to classical optimization. Bottou *et al.* survey large-scale optimization for ML and discuss tradeoffs among stochastic gradients and quasi-Newton methods under realistic compute budgets [3]. Adaptive scaling methods were motivated by sparse/heterogeneous gradients (AdaGrad) and led to Adam (moment estimates) [4], [7]. Subsequent work identified convergence issues and proposed variants such as AMSGrad [5]. AdamW clarified that "weight decay" is not always equivalent to an L_2 penalty under adaptive updates; decoupling weight decay often improves empirical behavior [6]. On the convex side, Boyd–Vandenberghe and Nocedal–Wright provide the theoretical backbone for convex duality/KKT systems and for quasi-Newton methods such as (L-)BFGS [1], [2]. For SVMs specifically, the primal/dual/KKT structure enables crisp geometric interpretation (margins, support vectors) [8].

Motivation and gap. Many optimizer comparisons focus only on deep networks (often lacking analytical anchors), while many LP/NLP treatments focus on convex problems that do not reflect modern AI training loops. This project bridges both by (i) expressing common ML objectives in an LP/NLP formulation, (ii) using convex tasks as "ground-truth" diagnostic benchmarks, and (iii) extending the same protocol to a small nonconvex neural network to observe where classical intuition still predicts behavior (and where it fails).

II. Problem Definition (Mathematical Formulation)

We study supervised learning with data $\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^d$ and labels y_i are either real-valued (regression) or binary $y_i \in \{-1, +1\}$.

A. Unified ERM + regularization (+ optional constraints)

We consider problems of the form

$$\min_{w\in\mathbb{R}^d} \; F(w) \equiv \frac{1}{n}\sum_{i=1}^{n} \ell(w; x_i, y_i) + \lambda R(w) \quad \text{s.t.} \quad g_j(w) \le 0,\; j = 1, \dots, m, \tag{1}$$

where w are parameters, ℓ is a loss (squared/logistic/hinge/cross-entropy), R is a regularizer (e.g., $R(w) = \frac{1}{2}\|w\|_2^2$ or $\|w\|_1$), $\lambda \ge 0$ is regularization strength, and g_j encode optional constraints (NLP component), e.g., a norm constraint $\|w\|_2 \le \tau$, monotonicity constraints, or proxy fairness constraints.

B. KKT conditions (interpretability/diagnostics)

Assuming differentiability and constraint qualification, KKT conditions for a solution $w^\star$ require multipliers $\mu^\star \in \mathbb{R}^m_{\ge 0}$ such that:

$$\begin{aligned} \text{Stationarity: } & \nabla F(w^\star) + \sum_{j=1}^{m} \mu_j^\star \nabla g_j(w^\star) = 0, \\ \text{Primal feasibility: } & g_j(w^\star) \le 0,\; j = 1, \dots, m, \\ \text{Dual feasibility: } & \mu_j^\star \ge 0,\; j = 1, \dots, m, \\ \text{Complementary slackness: } & \mu_j^\star g_j(w^\star) = 0,\; j = 1, \dots, m. \end{aligned} \tag{2}$$

These conditions help connect ML training to LP/NLP notions of constraints, penalties, and dual variables [1], [2].

C. Core convex benchmark problems (analytical anchors)

(A) Ridge regression (strongly convex, closed-form). Let $X \in \mathbb{R}^{n\times d}$, $y \in \mathbb{R}^n$. Solve

$$\min_{w} \; \frac{1}{2n}\|Xw - y\|_2^2 + \frac{\lambda}{2}\|w\|_2^2. \tag{3}$$

For $\lambda > 0$, the unique minimizer is

$$w^\star = (X^\top X + n\lambda I)^{-1} X^\top y, \tag{4}$$

providing a gold-standard reference for optimality gap and numerical error.

(B) Logistic regression with L_2 (convex, smooth). For $y_i \in \{-1, +1\}$,

$$\min_{w} \; \frac{1}{n}\sum_{i=1}^{n} \log\bigl(1 + \exp(-y_i x_i^\top w)\bigr) + \frac{\lambda}{2}\|w\|_2^2. \tag{5}$$

(C) Linear soft-margin SVM (convex, explicit constraints). Equivalent constrained form with slack $\xi \in \mathbb{R}^n_{\ge 0}$:

$$\min_{w,b,\xi\ge 0} \; \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n} \xi_i \quad \text{s.t.} \quad y_i(w^\top x_i + b) \ge 1 - \xi_i,\; \forall i. \tag{6}$$

This is ideal for illustrating KKT structure and the role of dual variables/support vectors [8].

D. Nonconvex extension (simulation benchmark)

(D) Small neural network (nonconvex). Let $f_\theta(\cdot)$ be a small MLP/CNN with parameters θ. We solve

$$\min_{\theta} \; \frac{1}{n}\sum_{i=1}^{n} \ell\bigl(f_\theta(x_i), y_i\bigr) + \lambda R(\theta), \tag{7}$$

typically with cross-entropy loss and weight decay/regularization. This tests how optimizer behavior changes once convex assumptions no longer apply.

E. What is the "problem to be solved" in this project?

The project objective is comparative and methodological:

> Given ML objectives spanning convex and non-convex regimes, determine how optimizer choice (first-order vs adaptive vs quasi-Newton, and optionally constrained methods) affects convergence, stability, and generalization under a standardized, reproducible protocol.

III. Methodology (Analytical + Numerical + Simulation)

A. Analytical component (LP/NLP concepts applied)

We will include course-aligned derivations and diagnostics:

- **Convexity/strong convexity:** implications for uniqueness and convergence in ridge/logistic/SVM [1].
- **Penalty vs constraint viewpoint:** relate regularization in (1) to constrained forms (e.g., $R(w)$ as a norm ball constraint) and discuss when they are equivalent [1], [2].
- **KKT/duality for SVM:** interpret complementary slackness and support vectors using (2) and (6) [8].
- **Conditioning/curvature:** explain why gradient methods can slow under ill-conditioning and why quasi-Newton methods (L-BFGS) can help on smooth convex objectives [2], [3].

B. Numerical component (optimizer benchmarking on convex tasks)

Optimizers (primary set).

- SGD (and full-batch GD where appropriate)
- SGD + Momentum (or Nesterov; we will pick one for scope control)
- Adam and AdamW (to highlight decoupled weight decay) [4], [6]
- L-BFGS as a quasi-Newton baseline [2]

Metrics. For each task, we will log:

- Objective value $F(w)$ vs epochs/iterations and vs wall-clock time
- Optimality gap where available:
  - Ridge: exact reference via (4)
  - Logistic/SVM: high-accuracy reference (tight tolerance L-BFGS or a trusted solver)

- Gradient norm $\|\nabla F(w)\|_2$ for smooth objectives (ridge/logistic)
- Robustness: sensitivity to learning rate, λ, and random seeds

Fair comparison protocol (reproducibility rules). To prevent "optimizer Olympics" (where the winner is whichever got more tuning love), we will standardize:

- Same data splits, preprocessing, initialization distributions, and random seeds
- Fixed compute budgets per method (e.g., same number of passes through data or same number of gradient evaluations), plus a shared stopping criterion (max epochs and/or tolerance on objective decrease)
- Equal hyperparameter tuning budgets (same-size grid or same number of trials per optimizer)
- Report mean ± standard deviation over multiple seeds

C. Simulation component (nonconvex neural network)

We will repeat the same logging protocol for a small network (kept intentionally small to maintain feasibility and interpretability):

- Dataset: MNIST (primary) or a small CIFAR-10 subset (secondary if time/compute allow)
- Outputs: training curves (loss/accuracy), test performance, generalization gap, seed-to-seed variability
- Focus point: compare Adam vs AdamW under matched "weight decay" values to illustrate the practical difference between an L_2 penalty and decoupled decay [6]

D. Optional constrained training add-on (if time permits)

To explicitly include a constrained NLP method beyond SVM:

- **Norm-constrained training:** $\|w\|_2 \leq \tau$ with projected gradient or penalty method
- (Stretch) A simple fairness proxy constraint in logistic regression (e.g., limit difference in predicted positive rates between two groups), treated via penalty or augmented-Lagrangian-style discussion [2]

E. Software tools

Planned stack:

- Python with NumPy/SciPy for convex objectives and linear algebra; SciPy L-BFGS(-B) for baselines
- PyTorch for neural networks and optimizer implementations (SGD/Momentum/Adam/AdamW)
- Matplotlib/Seaborn for plots; Pandas/CSV logging for tables; optional TensorBoard
- Reproducibility: fixed seeds, config files, and consistent environment capture

IV. Initial Results (Optional for Proposal Phase)

No finalized results are required at this stage. However, our first technical milestone will produce **sanity-check evidence** that the experimental harness is correct:

- Ridge regression: verify that iterative solvers converge to the closed-form $w^\star$ in (4), and report numerical error $\|w - w^\star\|$ and objective gap.
- Pilot comparison: produce at least one plot comparing (i) GD/SGD vs (ii) L-BFGS on ridge or logistic regression under matched stopping rules.

V. Expected Contributions / Deliverables

1) **Course-aligned review section** linking ML objectives to LP/NLP constructs: convexity, duality, KKT, conditioning, and quasi-Newton ideas.
2) **Reproducible benchmark suite** spanning convex (ridge/logistic/SVM) and nonconvex (small NN) tasks with a standardized protocol.
3) **Practitioner-oriented guidelines** ("if–then" rules) grounded in theory and observed behavior; e.g., when ill-conditioning favors curvature information (L-BFGS) vs when stochasticity/adaptation (AdamW) helps early training.

VI. Work Plan (Proposal-to-Report Path)

To keep scope realistic while aiming for a strong final report, we plan the following progression:

TABLE I
Planned milestones (high-level).

Milestone	Output
M1	Implement unified experiment harness; ridge closed-form verification.
M2	Convex benchmarks: ridge + logistic + SVM runs; optimizer sweeps; reference solutions.
M3	Nonconvex benchmark: small NN runs; Adam vs AdamW comparison; seed robustness.
M4	Write-up: theory (KKT/duality/conditioning) + results + guidelines; finalize reproducibility artifacts.

VII. References

References

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

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

[3] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.

[4] 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

[5] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of Adam and beyond," in *International Conference on Learning Representations (ICLR)*, 2018. [Online]. Available: https://arxiv.org/abs/1904.09237

[6] 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

[7] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011. [Online]. Available: http://jmlr.org/papers/v12/duchi11a.html

[8] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.