Learning to optimize with convergence guarantees

Andrea Martin

Abstract

The increasing reliance on numerical methods for controlling dynamical systems and training machine learning models underscores the need to devise algorithms that efficiently navigate complex optimization landscapes. Classical gradient descent methods offer strong theoretical worst-case guarantees for convex problems; however, they demand meticulous hyperparameter tuning for non-convex ones. The emerging paradigm of learning to optimize (L2O) automates the discovery of algorithms with optimized performance leveraging learning models and data – yet, it lacks a theoretical framework to analyze the convergence of the learned algorithms. In this extended abstract, we present results from [1] and we discuss related open research directions. Specifically, we present an unconstrained parametrization of all convergent algorithms for smooth non-convex objective functions. Notably, our framework is directly compatible with automatic differentiation tools, ensuring convergence by design while learning to optimize.

I. Introduction

Traditionally, optimization algorithms have been carefully designed by experts through case-by-case analyses of their convergence behavior when applied to problems with specific properties, such as smoothness and convexity. Interpreting iterative algorithms as evolving dynamical systems, [2] recently proposed an analysis and synthesis framework built on the notion of integral quadratic constraints from robust control theory, leading to the design of new methods with optimized convergence rates. To tackle non-convex optimization landscapes, the emerging paradigm of learning to optimize (L2O) embraces machine learning to automate the design of algorithms based on their performance on a set of training problems [3]. Despite outstanding empirical performance, however, learned optimizers generally lack convergence guarantees and require early-stopping or conservative fall-back mechanisms, raising concerns about reliability and generalization. In this extended abstract, we present the complete characterization of convergent algorithms for smooth non-convex objectives established in [1], which enables learning over all and only the set of convergent algorithms.

II. Problem Formulation

We focus on optimization problems of the form $x^\star = \arg\min_{x\in\mathbb{R}^d} \ f(x)$ where $f(\cdot)$ is bounded from below and has β-Lipschitz gradients, that is, $|\nabla f(x) - \nabla f(y)| \leq \beta|x-y|$ for all $x, y \in \mathbb{R}^d$. We denote the set of such β-smooth functions by $\mathcal{S}_\beta$. We describe an iterative optimization algorithm via the recursion

$$x_{t+1} = x_t + u_t = x_t + \pi_t(f, x_{t:0}) \,, \quad t \in \mathbb{N} \,, \tag{1}$$

where $x_0 \in \mathbb{R}^d$ is the initial guess, $x_t \in \mathbb{R}^d$ is the candidate solution vector after t iterations, and $u_t = \pi_t(f, x_{t:0}) \in \mathbb{R}^d$ is the algorithm update rule. We can write (1) compactly as $z\mathbf{x} = \mathbf{x} + \boldsymbol{\pi}(f, \mathbf{x}) + z\boldsymbol{\delta}^{x_0}$, where $\mathbf{x} = (x_0, x_1, \ldots)$, $z\mathbf{x} = (x_1, x_2, \ldots)$, $\boldsymbol{\delta}^{x_0} = (x_0, 0, \ldots) \in \ell_2$, and $\boldsymbol{\pi}(f, \cdot) = (\pi_0(f, x_0), \pi_1(f, x_{1:0}), \ldots)$ is a causal operator for any objective function $f(\cdot)$.

Definition 1: Consider the iteration (1). An update rule $\boldsymbol{\pi}(f, \mathbf{x})$ is *square-sum convergent* for f if for any $x_0 \in \mathbb{R}^d$

$$\|\boldsymbol{\pi}(f, \mathbf{x})\|^2 < \infty \,, \quad \|\nabla f(\mathbf{x})\|^2 < \infty \,. \tag{2}$$

We write $\boldsymbol{\pi}(f, \mathbf{x}) \in \Sigma(f)$ if $\boldsymbol{\pi}(f, \mathbf{x})$ is square-sum convergent for $f(\cdot)$. Given distributions $\mathcal{F}$ and $\mathcal{X}_0$ over functions in $\mathcal{S}_\beta$ and initial solutions $x_0 \in \mathbb{R}^d$, respectively, the problem of designing an optimal convergent algorithm becomes

$$\min_{\boldsymbol{\pi}} \ \mathbb{E}_{f\sim\mathcal{F}, x_0\sim\mathcal{X}_0} \left[\texttt{MetaLoss}(f, \mathbf{x})\right] \tag{3a}$$

$$\text{subject to } (1) \,, \quad \boldsymbol{\pi}(f, \mathbf{x}) \in \Sigma(f), \ \forall f \in \mathcal{S}_\beta \,, \tag{3b}$$

where, as suggested in [3], one can choose $\texttt{MetaLoss}(f, \mathbf{x}) = \sum_{t=0}^{T} \alpha_t |\nabla f(x_t)|^2 + \gamma_t f(x_t)$, with $\alpha_t \geq 0$ and $\gamma_t \geq 0$, to strike a balance between a rapid convergence to a stationary point and the quality of the corresponding solution.

III. Learning over all convergent algorithms

We now characterize update rules that converge according to Definition 1, and describe how to learn over them. We start by proving that if we perturb standard gradient descent with an ℓ_2 *"enhancement"* term – designed, e.g., to escape a bad local minimum or a saddle point – we preserve square-sum convergence to a critical point of f.

Lemma 1: For any $\mathbf{v} \in \ell_2$ and any $0 < \eta < \beta^{-1}$, the update rule given by

$$\boldsymbol{\pi}(f, \mathbf{x}) = -\eta \nabla f(\mathbf{x}) + \mathbf{v} \in \Sigma(f), \ \forall f \in \mathcal{S}_\beta \,. \tag{4}$$

The class of algorithms in the form (4) suggests a useful separation of roles; a gradient descent update can be used to ensure convergence, while an enhancement term $\mathbf{v} \in \ell_2$ can be learned to improve the algorithm performance. Nonetheless, a crucial question regarding the conservatism of searching over $\mathbf{v} \in \ell_2$ in (4) remains: *Can* any *convergent algorithm complying with* (3b) *be written as the sum of a gradient-based update and an enhancement signal* $\mathbf{v} \in \ell_2$ *as per* (4)*?*

In what follows, we answer in the affirmative by studying the closed-loop mappings induced by an update rule $\boldsymbol{\pi}(f, \mathbf{x})$.

Definition 2: For any update rule $\boldsymbol{\pi}(f, \mathbf{x})$, the mapping $(f, \boldsymbol{\delta}^{x_0}) \to (\mathbf{x}, \mathbf{u}, \nabla f(\mathbf{x}))$ is denoted as the *closed-loop mapping* induced by $\mathbf{u} = \boldsymbol{\pi}(f, \mathbf{x})$.

Lemma 2: For any $x_0 \in \mathbb{R}^d$ and $f \in \mathcal{S}_\beta$, let the closed-loop mapping induced by a policy $\mathbf{u} = \boldsymbol{\pi}(f, \mathbf{x})$ satisfying (3b) be

$$(f, \boldsymbol{\delta}^{x_0}) \to (\mathbf{x}_{\boldsymbol{\pi}}, \mathbf{u}_{\boldsymbol{\pi}}, \nabla f(\mathbf{x}_{\boldsymbol{\pi}})) \,. \tag{5}$$

Then, there exists $\mathbf{V} \in \mathcal{L}_2$ such that $(f, \boldsymbol{\delta}^{x_0}) \to (\mathbf{x}, -\eta \nabla f(\mathbf{x}) + \mathbf{V}(\boldsymbol{\delta}^{x_0}), \nabla f(\mathbf{x}))$ is equivalent to (5).

The completeness property stated above is key, as it implies that (4) encompasses all sum-square convergent algorithms – including those that *globally* minimize (3a). Together with Lemma 1, Lemma 2 leads to our main result.

Theorem 1: If $0 < \eta < \beta^{-1}$, the meta-optimization problem (3) is equivalent to

$$\min_{\mathbf{V} \in \mathcal{L}_2} \quad \mathbb{E}_{f \sim \mathcal{F}, x_0 \sim \mathcal{X}_0} \left[\texttt{MetaLoss}(f, \mathbf{x}) \right] \tag{6a}$$

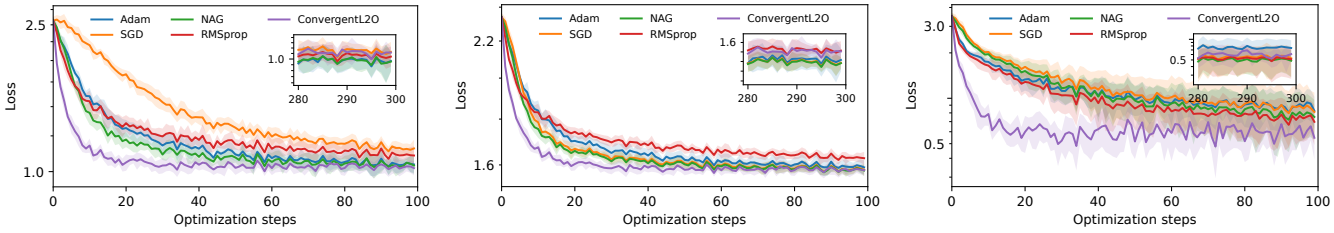
$$\text{subject to} \quad z\mathbf{x} = \mathbf{x} - \eta \nabla f(\mathbf{x}) + \mathbf{V}(\boldsymbol{\delta}^{x_0}) + z\boldsymbol{\delta}^{x_0} \,. \tag{6b}$$

A few comments are in order. First, any possibly suboptimal solution $\mathbf{V} \in \mathcal{L}_2$ to (6) yields a converging algorithm complying with (3b). Second, every converging algorithm complying with (3b) is recovered by appropriately choosing $\mathbf{V} \in \mathcal{L}_2$ with no conservatism. Third, as the convergence constraint (3b) simplifies to $\mathbf{V} \in \mathcal{L}_2$, we can use finite-dimensional approximations of operators in $\mathcal{L}_2$ available in the literature to translate (6) into the unconstrained optimization problem of learning the best parameter $\theta \in \mathbb{R}^D$ via automatic differentiation tools.

Remark 1: While Theorem 1 proves that designing an update rule that solely reacts to $x_0 \sim \mathcal{X}_0$ is sufficient for achieving meta-optimal behaviors, introducing explicit dependence of $\mathbf{V}$ on additional input features could significantly improve how effectively we navigate the meta-optimization landscape. For instance, by defining $\boldsymbol{\omega} = \boldsymbol{\Omega}(\mathbf{x}, \nabla f(\mathbf{x}), f(\mathbf{x}))$ and $\mathbf{z} = \mathbf{Z}(\boldsymbol{\delta}^{x_0})$, where $\boldsymbol{\Omega} : \ell \to \ell$ and $\mathbf{Z} \in \mathcal{L}_2$ are operators to be freely designed, a simple approach to preserve sum-square convergence while introducing input features is to consider $\mathbf{v} \in \ell_2$ parametrized as $v_t = |z_t||\omega_t|^{-1}\omega_t$.

IV. Experiments

In Figure 1, we compare the performance of our `ConvergentL2O` algorithm and that of classical fine-tuned optimizers in training a neural network for image classification on the MNIST dataset. Remarkably, despite being trained to optimize the parameters of a neural network with $\tanh$, our algorithm also generalizes to structurally different activation functions.



(a) Activation function: $\tanh$. (b) Activation function: $\mathrm{sigmoid}$. (c) Activation function: ReLU.

Fig. 1. Training curves of learned and hand-crafted optimizers; shaded areas and solid lines denote standard deviations and mean values, respectively.

V. Conclusion

We have presented a methodology for learning over all convergent update rules for smooth non-convex optimization. By synergizing systems theory with the emerging L2O paradigm, we aimed to close the gap between theory-based algorithm design and example-driven approaches that are the hallmark of machine learning. Building on the proposed control-theoretic perspective we have adopted and on the formalism of fixed-point operators, our ongoing work [4] studies how to enhance the performance of legacy convex optimization algorithms, such as Nesterov's accelerated gradient method, on a class of relevant problems, while preserving stronger linear convergence guarantees. We foresee that the formalism of fixed-point operators will also play a central role in extending our approach to game-theoretic settings, where agents aim to rapidly converge to equilibria that appropriately balance fairness and social welfare. Further avenues for future research include conducting a formal generalization analysis and studying online and constrained optimization scenarios.

References

[1] A. Martin and L. Furieri, "Learning to optimize with convergence guarantees using nonlinear system theory," *IEEE Control Systems Letters*, vol. 8, pp. 1355–1360, 2024.

[2] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 57–95, 2016.

[3] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," *Advances in neural information processing systems*, vol. 29, 2016.

[4] A. Martin, I. R. Manchester, and L. Furieri, "A characterization of linearly convergent algorithms in convex optimization," *In preparation*, 2025.