Towards Constituting Mathematical Structures for Learning to Optimize

Jialin Liu* (Alibaba) Xiaohan Chen* (Alibaba) Zhangyang Wang (UT Austin) Wotao Yin (Alibaba) HanQin Cai (UCF)

Overview

A generic learning-to-optimize (L2O) approach parameterizes the iterative update rule and learns the update direction as a black-box network. While the generic approach is widely applicable, the learned model can overfit and may not generalize well to out-of-distribution test sets.

We derive the basic mathematical conditions that successful update rules commonly satisfy. Consequently, we propose a novel L2O model with a mathematics-inspired structure that is broadly applicable and generalized well to out-of-distribution problems. [1]

arXiv *GitHub*

Introduction

In this study, we consider optimization problems in the form of

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} F(\boldsymbol{x}) = f(\boldsymbol{x}) + r(\boldsymbol{x}),$$

where $f(\boldsymbol{x})$ is a smooth convex function with Lipschitz continuous gradient, and $r(\boldsymbol{x})$ is a convex function that may be non-smooth.

Generic update. A general parameterized update rule is written as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{d}_k(\boldsymbol{z}_k;\phi), \tag{1}$$

where $\boldsymbol{z}_k \in \mathcal{Z}$ is the *input vector* and $\mathcal{Z}$ is the *input space*. The input vector may involve dynamic information such as $\{\boldsymbol{x}_k, F(\boldsymbol{x}_k), \nabla F(\boldsymbol{x}_k)\}$. For example in [2], the input vector is $\boldsymbol{z}_k = [\boldsymbol{x}_k^\top, \nabla F(\boldsymbol{x}_k)^\top]^\top$ with the input space being $\mathcal{Z} = \mathbb{R}^{2n}$, and the update $\boldsymbol{d}_k$ is generated using an LSTM network parameterized by ϕ and shared across coordinates of $\boldsymbol{x}_k$.

Definition 1 (Spaces of Objective Functions) *We define function spaces $\mathcal{F}(\mathbb{R}^n)$ and $\mathcal{F}_L(\mathbb{R}^n)$ as*

$$\mathcal{F}(\mathbb{R}^n) = \Big\{ r:\mathbb{R}^n \to \mathbb{R} \,\Big|\, r \text{ is proper, closed and convex} \Big\},$$

$$\mathcal{F}_L(\mathbb{R}^n) = \Big\{ f:\mathbb{R}^n \to \mathbb{R} \,\Big|\, f \text{ is convex, differentiable, and } \|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \le L\|\boldsymbol{x}-\boldsymbol{y}\|, \forall \boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^n \Big\}.$$

Definition 2 (Space of Update Rules) *Let $\mathrm{J}\boldsymbol{d}(\boldsymbol{z})$ denote the Jacobian matrix of operator $\boldsymbol{d}:\mathcal{Z}\to\mathbb{R}^n$ and $\|\cdot\|_{\mathrm{F}}$ denote Frobenius norm, we define the space:*

$$\mathcal{D}_C(\mathcal{Z}) = \Big\{ \boldsymbol{d}:\mathcal{Z}\to\mathbb{R}^n \,\Big|\, \boldsymbol{d} \text{ is differentiable, } \|\mathrm{J}\boldsymbol{d}(\boldsymbol{z})\|_{\mathrm{F}} \le C,\ \forall \boldsymbol{z}\in\mathcal{Z} \Big\}.$$

References

[1] J. Liu, X. Chen, Z. Wang, W. Yin, and H. Cai, "Towards constituting mathematical structures for learning to optimize," in *ICML*, 2023.

[2] M. Andrychowicz, M. Denil, S. Gomez, *et al.*, "Learning to learn by gradient descent by gradient descent," *Advances in neural information processing systems*, 2016.

[3] D. Dua and C. Graff, *UCI machine learning repository*, 2017.

Main Results

We use explicit formula for f and implicit formula for r:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{d}_k(\boldsymbol{x}_k, \nabla f(\boldsymbol{x}_k), \boldsymbol{x}_{k+1}, \boldsymbol{g}_{k+1}), \tag{2}$$

where $\boldsymbol{g}_{k+1} \in \partial r(\boldsymbol{x}_{k+1})$ and $\boldsymbol{z}_k = [\boldsymbol{x}_k^\top, \nabla f(\boldsymbol{x}_k)^\top, \boldsymbol{x}_{k+1}^\top, \boldsymbol{g}_{k+1}^\top]^\top$ as in (1) and input space is $\mathcal{Z} = \mathbb{R}^{4n}$.

The convexity of f and r implies that $\mathbf{0} \in \nabla f(\boldsymbol{x}_*) + \partial r(\boldsymbol{x}_*)$ if and only if $\boldsymbol{x}_* \in \operatorname{argmin}_{\boldsymbol{x}} F(\boldsymbol{x})$. Thus, it holds that $-\nabla f(\boldsymbol{x}_*) \in \partial r(\boldsymbol{x}_*)$. With $\boldsymbol{g}_* = -\nabla f(\boldsymbol{x}_*)$, we can write the following two conditions

Asymptotic fixed point condition (FP3). For any $\boldsymbol{x}_* \in \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} F(\boldsymbol{x})$, it holds that $\lim_{k\to\infty} \boldsymbol{d}_k(\boldsymbol{x}_*, \nabla f(\boldsymbol{x}_*), \boldsymbol{x}_*, -\nabla f(\boldsymbol{x}_*)) = \mathbf{0}$.

Global convergence (GC3). For any sequences $\{\boldsymbol{x}_k\}_{k=0}^\infty$ generated by (2), there exists $\boldsymbol{x}_* \in \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} F(\boldsymbol{x})$ such that $\lim_{k\to\infty} \boldsymbol{x}_k = \boldsymbol{x}_*$.

Theorem 3 *Given $f \in \mathcal{F}_L(\mathbb{R}^n)$ and $r \in \mathcal{F}(\mathbb{R}^n)$, we pick a sequence of operators $\{\boldsymbol{d}_k\}_{k=0}^\infty$ with $\boldsymbol{d}_k \in \mathcal{D}_C(\mathbb{R}^{4n})$ and generate $\{\boldsymbol{x}_k\}_{k=0}^\infty$ by (2). If both (FP3) and (GC3) conditions hold, then for all $k = 0,1,2,\cdots$, there exist $\mathbf{P}_k \in \mathbb{R}^{n\times n}$ and $\boldsymbol{b}_k \in \mathbb{R}^n$ satisfying*

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \mathbf{P}_k(\nabla f(\boldsymbol{x}_k) - \boldsymbol{g}_{k+1}) - \boldsymbol{b}_k,\ \boldsymbol{g}_{k+1} \in \partial r(\boldsymbol{x}_{k+1}),$$

with $\mathbf{P}_k$ is bounded and $\boldsymbol{b}_k \to \mathbf{0}$ as $k \to \infty$. If we further assume $\mathbf{P}_k$ is symmetric positive definite, then $\boldsymbol{x}_{k+1}$ is uniquely determined given $\boldsymbol{x}_k$ through

$$\boldsymbol{x}_{k+1} = \operatorname{prox}_{r,\mathbf{P}_k}(\boldsymbol{x}_k - \mathbf{P}_k \nabla f(\boldsymbol{x}_k) - \boldsymbol{b}_k). \tag{3}$$

Longer horizon. Introduce an auxiliary variable $\boldsymbol{y}_k$ that encodes historical information through operator $\boldsymbol{y}_k = \boldsymbol{m}(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}, \cdots, \boldsymbol{x}_{k-T})$, leading to the extended update rule and conditions. With $\boldsymbol{g}_{k+1} \in \partial r(\boldsymbol{x}_{k+1})$,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{d}_k(\boldsymbol{x}_k, \nabla f(\boldsymbol{x}_k), \boldsymbol{x}_{k+1}, \boldsymbol{g}_{k+1}, \boldsymbol{y}_k, \nabla f(\boldsymbol{y}_k)). \tag{4}$$

(FP4) For any $\boldsymbol{x}_* \in \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} F(\boldsymbol{x})$, it holds that $\boldsymbol{m}(\boldsymbol{x}_*, \boldsymbol{x}_*, \cdots, \boldsymbol{x}_*) = \boldsymbol{x}_*$ and $\lim_{k\to\infty} \boldsymbol{d}_k(\boldsymbol{x}_*, \nabla f(\boldsymbol{x}_*), \boldsymbol{x}_*, -\nabla f(\boldsymbol{x}_*), \boldsymbol{x}_*, \nabla f(\boldsymbol{x}_*)) = \mathbf{0}$.

(GC4) For any sequences $\{\boldsymbol{x}_k, \boldsymbol{y}_k\}_{k=0}^\infty$ generated by (4), there exists one $\boldsymbol{x}_* \in \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} F(\boldsymbol{x})$ such that $\lim_{k\to\infty} \boldsymbol{x}_k = \lim_{k\to\infty} \boldsymbol{y}_k = \boldsymbol{x}_*$.

Theorem 4 *Suppose $T = 1$. Given $f \in \mathcal{F}_L(\mathbb{R}^n)$ and $r \in \mathcal{F}(\mathbb{R}^n)$, we pick an operator $\boldsymbol{m} \in \mathcal{D}_C(\mathbb{R}^{2n})$ and a sequence of operators $\{\boldsymbol{d}_k\}_{k=0}^\infty$ with $\boldsymbol{d}_k \in \mathcal{D}_C(\mathbb{R}^{6n})$. If both (FP4) and (GC4) hold, for any bounded matrix sequence $\{\mathbf{B}_k\}_{k=0}^\infty$, there exist $\mathbf{P}_{1,k}, \mathbf{P}_{2,k}, \mathbf{A}_k \in \mathbb{R}^{n\times n}$ and $\boldsymbol{b}_{1,k}, \boldsymbol{b}_{2,k} \in \mathbb{R}^n$ satisfying*

$$\begin{aligned} \boldsymbol{x}_{k+1} = \boldsymbol{x}_k &- (\mathbf{P}_{1,k} - \mathbf{P}_{2,k})\nabla f(\boldsymbol{x}_k) - \mathbf{P}_{2,k}\nabla f(\boldsymbol{y}_k) - \boldsymbol{b}_{1,k} && (5)\\ &- \mathbf{P}_{1,k}\boldsymbol{g}_{k+1} - \mathbf{B}_k(\boldsymbol{y}_k - \boldsymbol{x}_k),\ \boldsymbol{g}_{k+1} \in \partial r(\boldsymbol{x}_{k+1}), \\ \boldsymbol{y}_{k+1} = (\mathbf{I} &- \mathbf{A}_k)\boldsymbol{x}_{k+1} + \mathbf{A}_k\boldsymbol{x}_k + \boldsymbol{b}_{2,k} && (6) \end{aligned}$$

for all $k = 0,1,2,\cdots$, with $\{\mathbf{P}_{1,k}, \mathbf{P}_{2,k}, \mathbf{A}_k\}$ bounded and $\boldsymbol{b}_{1,k} \to \mathbf{0}, \boldsymbol{b}_{2,k} \to \mathbf{0}$ as $k \to \infty$. If we further assume $\mathbf{P}_{1,k}$ is uniformly symmetric positive definite, then we can substitute $\mathbf{P}_{2,k}\mathbf{P}_{1,k}^{-1}$ with $\mathbf{B}_k$ and obtain

$$\begin{aligned} \hat{\boldsymbol{x}}_k &= \boldsymbol{x}_k - \mathbf{P}_{1,k}\nabla f(\boldsymbol{x}_k), \quad \hat{\boldsymbol{y}}_k = \boldsymbol{y}_k - \mathbf{P}_{1,k}\nabla f(\boldsymbol{y}_k), \\ \boldsymbol{x}_{k+1} &= \operatorname{prox}_{r,\mathbf{P}_{1,k}}\Big((\mathbf{I} - \mathbf{B}_k)\hat{\boldsymbol{x}}_k + \mathbf{B}_k\hat{\boldsymbol{y}}_k - \boldsymbol{b}_{1,k} \Big), \qquad (7) \\ \boldsymbol{y}_{k+1} &= \boldsymbol{x}_{k+1} + \mathbf{A}_k(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \boldsymbol{b}_{2,k}. \end{aligned}$$

Numerical Validation

LSTM Parameterization. We choose diagonal $\mathbf{P}_{1,k}, \mathbf{B}_k, \mathbf{A}_k$ over full matrices for efficiency. Similar to [2], we model $\boldsymbol{p}_k$, $\boldsymbol{a}_k$, $\boldsymbol{b}_k$, $\boldsymbol{b}_{1,k}$, $\boldsymbol{b}_{2,k}$ as the output of a coordinate-wise LSTM, which is parameterized by learnable parameters ϕ_{LSTM} and takes the current estimate $\boldsymbol{x}_k$ and the gradient $\nabla f(\boldsymbol{x}_k)$ as the input:

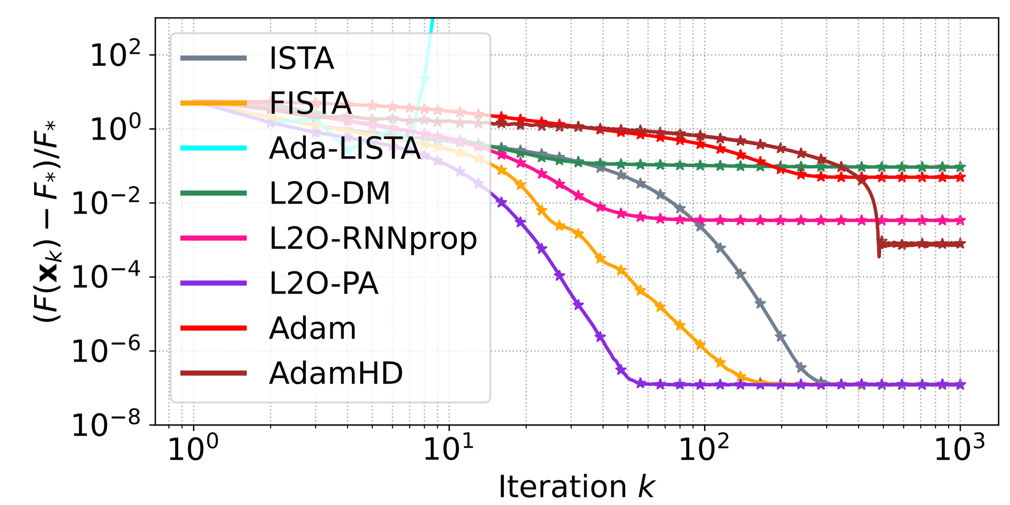
$$\begin{aligned} \boldsymbol{o}_k, \boldsymbol{h}_k &= \mathrm{LSTM}(\boldsymbol{x}_k, \nabla f(\boldsymbol{x}_k), \boldsymbol{h}_{k-1}; \phi_{\mathrm{LSTM}}), \\ \boldsymbol{p}_k, \boldsymbol{a}_k, \boldsymbol{b}_k, \boldsymbol{b}_{1,k}, \boldsymbol{b}_{2,k} &= \mathrm{MLP}(\boldsymbol{o}_k; \phi_{\mathrm{MLP}}). \end{aligned} \tag{8}$$

Here, $\boldsymbol{h}_k$ is the internal state maintained by the LSTM with $\boldsymbol{h}_0$ randomly sampled from Gaussian distribution.

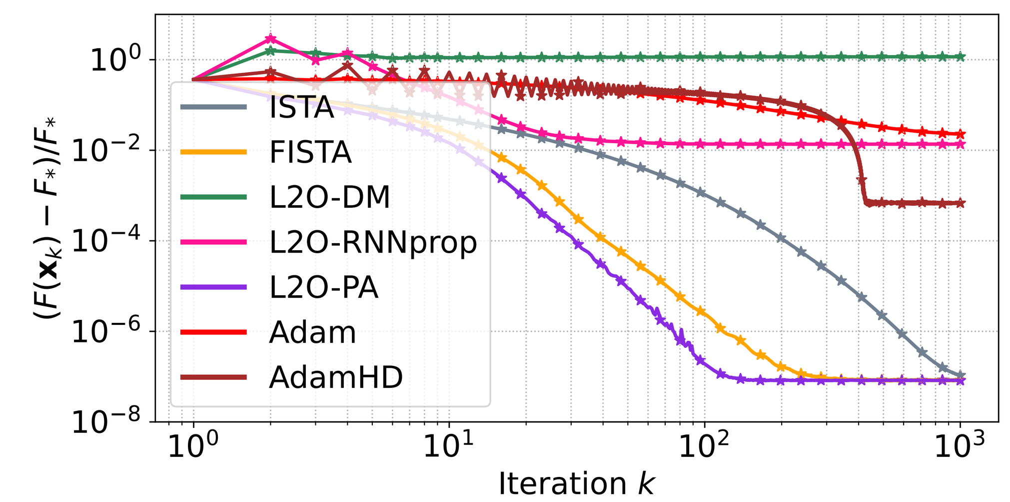
Experiment Settings. We validate our theories with experiments on LASSO and logistic regression using both synthetic data and real data.

- For our method, we learn to predict the diagonal $\boldsymbol{p}_k$ and $\boldsymbol{a}_k$ with LSTM.
- For LASSO, we sample $\boldsymbol{A} \in \mathbb{R}^{250\times 500}$, $\boldsymbol{b} \in \mathbb{R}^{250}$ for the synthetic setting; $\boldsymbol{A} \in \mathbb{R}^{64\times 128}$, $\boldsymbol{b} \in \mathbb{R}^{64}$ extracted with 1,000 8×8 patches from BSD500.
- For logistic regression, we sample $\boldsymbol{A} \in \mathbb{R}^{1000\times 50}$ for the synthetic setting and use *Ionosphere* and *Spambase* datasets as real data [3].
- Models trained on synthetic data are applied to real data directly.

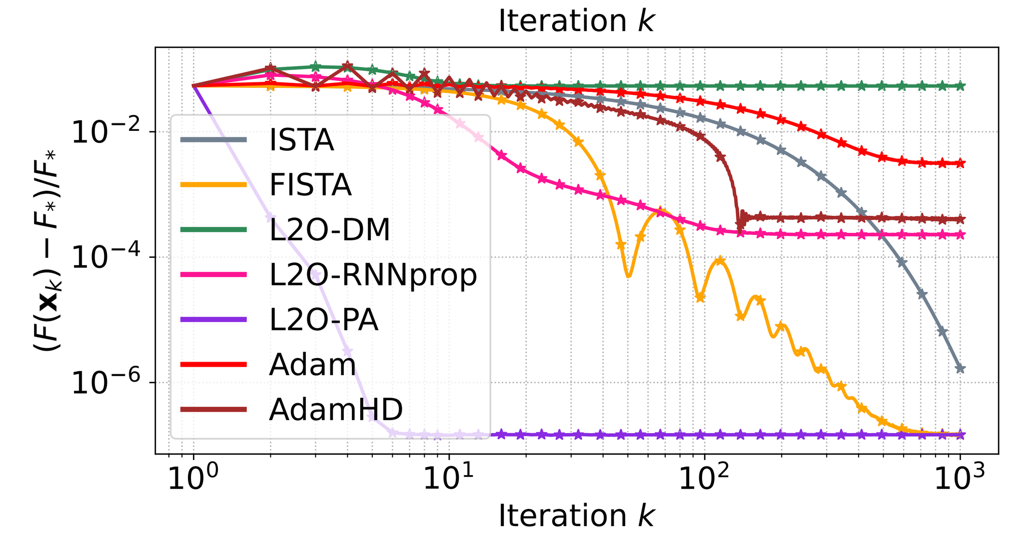
LASSO Synthetic



LASSO Real *Directly Transferred from Synthetic*



Logistic Synthetic



Logistic Ionosphere *Directly Transferred from Synthetic*