MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Learning a Constrained Optimizer: A Primal Method

Liu, Tao; Cherian, Anoop

TR2023-003 January 24, 2023

Abstract

There has been significant interest in developing methods that bridge between classical optimization and modern deep learning, under the broad theme of learning to optimize (L2O), for improved optimizers. In this paper, we propose Switch-L2O – a new primal-only method for learning a constraint optimizer. Empirically, our method is shown to enjoy a better optimality gap and reduces constraint violations against prior methods on convex and nonconvex optimization problems with possibly nonconvex constraints.

AAAI Conference on Artificial Intelligence 2023

© 2023 AAAI. Personal use of this material is permitted. Permission from AAAI must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Learning a Constrained Optimizer: A Primal Method

Tao Liu* and Anoop Cherian†

Abstract

There has been significant interest in developing methods that bridge between classical optimization and modern deep learning, under the broad theme of learning to optimize (L2O), for improved optimizers. In this paper, we propose Switch-L2O – a new primal-only method for learning a constraint optimizer. Empirically, our method is shown to enjoy a better optimality gap and reduces constraint violations against prior methods on convex and nonconvex optimization problems with possibly nonconvex constraints.

1 Introduction

Learning to optimize (L2O) is an emerging approach that leverages machine learning to develop optimization methods, aiming at faster convergence as well as higher accuracy [3]. The learned optimizer is trained over a set of similar optimizees that represent the task distribution (offline training) and designed to solve unseen optimizees from the same distribution (online testing). The goal is to minimize the weighted sum of the objective function $f(x_t)$ over a time span (i.e., unrolling length) T given by

$$\min_{\phi} \mathbb{E}_{f\in\mathcal{T}}\left[\sum_{t=1}^{T} w_t f(x_t)\right], \quad \text{with } x_{t+1} = x_t - g(z_t;\phi), t = 1,\ldots,T-1, \tag{1}$$

where the mapping function g is parameterized by ϕ and $\mathcal{T}$ represents the target task distribution, with w_t set to 1.

Since naive L2O approaches cannot enforce the hard constraints, Donti et al. [4] designed deep constraint completion and correction (DC3), which first approximately solves for a partial set of variables via neural networks, then completes other variables using the equality constraints, and finally fixes inequality violations by multi-step gradient-based updates. Shen et al. [5] proposed Twin-L2O, using two LSTMs to separately update decision variables and dual variables. Although the above methods achieve some success, they suffer from several disadvantages, namely: 1) the relative error of objective and the relative error of decision variables are nontrivial, 2) the results of DC3 are sensitive to the fixed dual variable and DC3 requires tuning the dual variable for different problems (and thus not general), 3) gradient descent in the inequality correction step of DC3 may not work for general optimization formulations when the initialization is not close to an optimum, 4) the training of Twin-L2O may be unstable, and 5) the primal-dual framework of Twin-L2O is sensitive to the initialization of the Lagrange multiplier. To circumvent these issues, we propose **Switch-L2O – a model-free primal-only L2O method designed for general constrained optimization** (possibly nonconvex) with a small constraint violation and optimality gap.

*Texas A&M University, College Station, TX email: tliu@tamu.edu
†Mitsubishi Electric Research Labs, Cambridge, MA email: cherian@merl.com

2 Switch-L2O

Since the equality constraints can be written as two inequality constraints, we consider the general constrained optimization setting as follows:

$$\underset{y\in\mathbb{R}^n}{\text{minimize}}\ f(y), \quad \text{s.t. } g_i(y)\le 0, \forall i\in[m], \tag{2}$$

where f and g are potentially nonlinear and nonconvex. We will use g (without suffix) to denote the set of all g_i.

2.1 Primal-dual framework

As is well-known, general constrained optimization formulation, when solved by the primal-dual framework, is equivalent to handling a minimax optimization with dual variables λ satisfying $\lambda\ge 0$. That is,

$$\min_{y\in\mathbb{R}^n}\max_{\lambda\in\mathbb{R}^m_+}\ f(y)+\lambda^T g(y) =: L(y,\lambda). \tag{3}$$

This formulation is used in Twin-L2O [5], proposing two LSTMs that separately update y and λ as:

$$y_{t+1}=y_t+\Delta y_t,\ \text{where}\ \left(\Delta y_t, h_{t+1}^{\min}\right)=\text{LSTM}_{\min}\left(\left[\nabla_y L\left(y_t,\lambda_t\right),\nabla_\lambda L\left(y_t,\lambda_t\right)\right],h_t^{\min};\phi^{\min}\right),$$
$$\lambda_{t+1}=\left[\lambda_t+\Delta\lambda_t\right]_+,\ \text{where}\ \left(\Delta\lambda_t, h_{t+1}^{\max}\right)=\text{LSTM}_{\max}\left(\left[\nabla_y L\left(y_{t+1},\lambda_t\right),\nabla_\lambda L\left(y_{t+1},\lambda_t\right)\right],h_t^{\max};\phi^{\max}\right),$$

where $[\cdot]_+=\max(\cdot,0)$. The corresponding L2O loss is given by:

$$\mathcal{L}\left(\phi^{\min},\phi^{\max}\right)=\mathbb{E}_{L\sim\mathcal{T}}\left[\sum_{t=1}^{T}\left\{\left(L\left(y_t,\lambda_{t-1}\right)-L\left(y_t,\lambda_t\right)\right)+\left(L\left(y_t,\lambda_{t-1}\right)-L\left(y_{t-1},\lambda_{t-1}\right)\right)\right\}\right]. \tag{4}$$

2.2 New primal-only framework

As alluded to above, in practice a primal-dual optimization framework may be sensitive to the initialization of the Lagrange multiplier [1]. In the proposed Switch-L2O, we use only the primal form in the setting of learning a constrained optimizer alternating the optimization between the objective and the constraints using a single LSTM. The loss function at each time step t is defined as:

$$\mathcal{L}_t(y_t;\phi)=\begin{cases} f(y_t), & \text{if } g_i(y_t)\le\xi,\forall i\in[m]\\ \sum_{i=1}^m(\max(g_i(y_t),0))^2, & \text{if } g_i(y_t)>\xi,\exists i\in[m].\end{cases} \tag{5}$$

where $\xi\ge 0$ is a slack hyperparameter and the loss function $\mathcal{L}(\phi)$ (to learn the optimizer) is defined as

$$\mathcal{L}(\phi)=\mathbb{E}_{(f,g)\sim\mathcal{T}}\left[\sum_{t=1}^{T}\mathcal{L}_t(y_t;\phi)\right], \tag{6}$$

$$\text{with the update:}\quad y_{t+1}=y_t+\Delta y_t \text{ and } (\Delta y_t,h_{t+1})=\text{LSTM}\left(\nabla_y\mathcal{L}_t(y_t;\phi),h_t;\phi\right). \tag{7}$$

3 Experimental results

We adopt three metrics to analyze the performance of the new primal framework. The first one is the relative error of objective (average over all test samples), i.e., $\frac{1}{n_{test}}\sum_{i=1}^{n_{test}}\frac{f^i(y_t^i)-f^i(y_*^i)}{|f^i(y_*^i)|}$. The second one is the mean of constraint violation (average), i.e., $\frac{1}{n_{test}}\sum_{i=1}^{n_{test}}\frac{1}{m}\sum_{j=1}^{m}\max(g_j^i(y_t^i),0)$. The third one is the relative error of decision variables (average), i.e., $\frac{1}{n_{test}}\sum_{i=1}^{n_{test}}\frac{\|y_t^i-y_*^i\|}{\|y_*^i\|}$. We use OSQP [6] to calculate y_*^i for convex optimization, while IPOPT solver [7] finds y_*^i for nonconvex optimization. While conducting our experiments, we found that the Twin-L2O baseline model diverges when we only use one of the gradients as input in LSTM_{Min} and LSTM_{Max} even if the dimension of decision variables is 1. To use both gradients as input, we should guarantee the dimension of y and λ are the same, which can be achieved by duplicating the provided inequalities.

Table 1: Switch-L20 results on a convex QP task.

Method	Rel-obj	Vio-mean	Rel-d.v.s
DC3 ($\lambda = 1$)	-0.0592	0.0212	0.2128
DC3 ($\lambda = 5$)	-0.0940	0.0039	0.2637
Twin-L2O	-0.3171	0.0779	0.6423
Switch-L2O ($\xi = 0.1$)	-0.1526	**0.0031**	0.4936
Switch-L2O ($\xi = 1$)	**-0.0015**	0.1147	**0.1482**

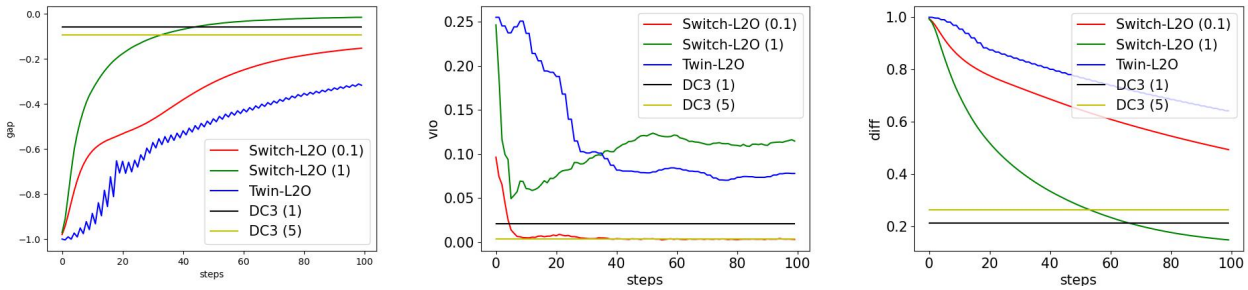

Figure 1: Switch-L20 Results on a convex QP task. In brackets, we show the slack parameter, ξ.

3.1 Convex QP

To test the effectiveness of the proposed primal framework, we consider a convex quadratic programming (QP) problem with 100-dimensional decision variables and 50 inequality constraints, given by:

$$\underset{y\in\mathbb{R}^n}{\text{minimize}}\ \frac{1}{2}y^TQy + p^Ty, \quad \text{s.t. } Ay \le x, \tag{8}$$

for constants $Q \in \mathbb{R}^{n\times n} \succeq 0, p \in \mathbb{R}^n, A \in \mathbb{R}^{n_{\text{ineq}}\times n}$, and variable $x \in \mathbb{R}^{n_{\text{ineq}}}$. We take Q to be a diagonal matrix with all diagonal entries drawn i.i.d. from the uniform distribution on $[0, 1]$, p with entries drawn i.i.d. from the uniform distribution on $[0, 1]$, x with entries drawn i.i.d. from the uniform distribution on $[-1, 1]$, and A with entries drawn i.i.d. from the standard normal distribution.

The vast majority of hyperparameter selection follows [2]. We use 200 optimizee instances for training, 100 optimizees for validation, and 100 hold-out instances for testing. We train Switch-L2O solvers for 200 epochs, using Adam with a constant learning rate 0.003. For each epoch, Switch-L2O will update the optimizee parameters for 100 iterations, with its unrolling length $T = 20$.

Table 1 and Figure 1 illustrate the superior performance of Switch-L2O in solving convex QP tasks, which enjoys a smaller optimality gap and constraint violation compared with DC3 and Twin-L2O. For Switch-L2O itself, there is a trade-off between the optimality gap and constraint violation by adjusting the slack parameter ξ. With the increase of the slack parameter (from 0.1 to 1.0), the relative error of objective and decision variables decreases at the cost of an increase in constraint violation. Additionally, a trade-off exists for the DC3 method by selecting different fixed dual variables λ. A smaller dual variable ($\lambda = 1$) contributes to a smaller optimality gap, while it suffers from a larger violation.

3.2 Nonconvex optimization

We also empirically evaluate our approach on two nonconvex formulations, namely: (i) nonconvex objective with linear constraints (9), and (ii) nonconvex objective with nonconvex constraints (10):

$$\underset{y\in\mathbb{R}^n}{\text{minimize}}\ \frac{1}{2}y^TQy + p^T\sin(y), \quad \text{s.t. } Ay \le x. \tag{9}$$

$$\underset{y\in\mathbb{R}^n}{\text{minimize}}\ \frac{1}{2}y^TQy + p^T\sin(y), \quad \text{s.t. } A\sin(y) \le x. \tag{10}$$

Table 2: Switch-L20 results on nonconvex optimization.

Method	Nonconvex obj & linear constraints			Nonconvex obj & constraints		
	Rel-obj	Vio-mean	Rel-d.v.s	Rel-obj	Vio-mean	Rel-d.v.s
DC3 ($\lambda = 1$)	-0.0977	0.0201	0.2625	-0.0836	0.0620	0.2432
Twin-L2O	-0.0614	0.0641	0.274	-0.1632	0.0019	0.8134
Switch-L2O ($\xi = 0.1$)	-0.0353	**0.0005**	0.2148	**-0.0257**	**0.0070**	0.1771
Switch-L2O ($\xi = 1$)	**0.0251**	0.1970	**0.0816**	0.0341	0.2015	**0.0781**

For nonconvex optimization, Table 2 and Figures 2 and 3 display the superior performance of Switch-L2O, which enjoys a smaller optimality gap and constraint violation compared with DC3 and Twin-L2O. Similar to convex optimization, a trade-off between the optimality gap and constraint violation is made by tuning the slack parameter ξ for Switch-L2O.

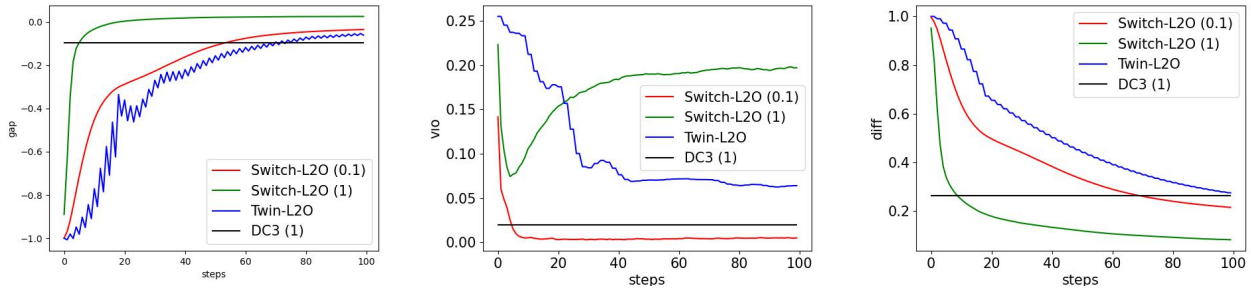

Figure 2: Results on nonconvex optimization (nonconvex objective with linear constraints).

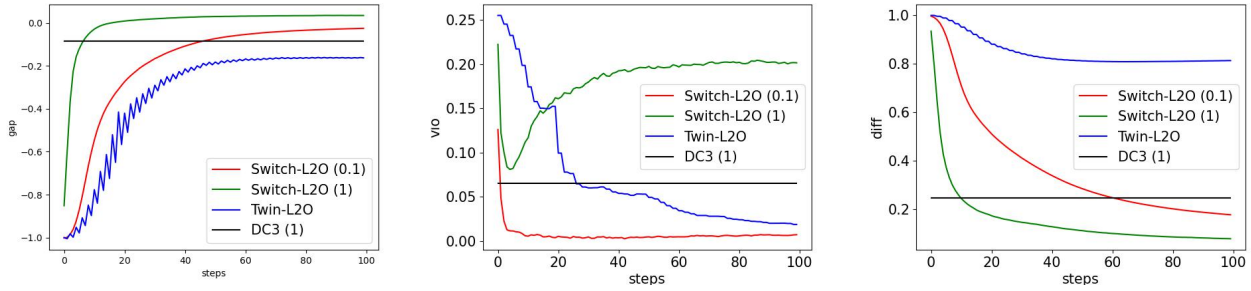

Figure 3: Results on a nonconvex optimization (nonconvex objective and nonconvex constraints).

4 Conclusions and future work

In this paper, we present Switch-L2O, a general-purpose constrained optimization solver trained to optimize using neural networks. Our key idea is to split the optimization iterations either to minimize the objectives or the constraint violations. The former learns to optimize the objective, while the latter ensures the solutions are within the feasible region. Our formulation is primal-only and thus avoids the drawbacks of prior approaches, while also empirically demonstrating better convergences and optimality gaps. A future direction of this work would be to improve the computational cost of our approach.

References

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In International conference on machine learning, pages 22–31. PMLR, 2017.

[2] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. Advances in neural information processing systems, 29, 2016.

[3] Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. arXiv preprint arXiv:2103.12828, 2021.

[4] Priya L Donti, David Rolnick, and J Zico Kolter. Dc3: A learning method for optimization with hard constraints. arXiv preprint arXiv:2104.12225, 2021.

[5] Jiayi Shen, Xiaohan Chen, Howard Heaton, Tianlong Chen, Jialin Liu, Wotao Yin, and Zhangyang Wang. Learning a minimax optimizer: A pilot study. In International Conference on Learning Representations, 2020.

[6] Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd. Osqp: An operator splitting solver for quadratic programs. Mathematical Programming Computation, 12(4):637–672, 2020.

[7] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical programming, 106(1):25–57, 2006.