Towards Constituting Mathematical Structures for Learning to Optimize

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

ICML 2023

Learning to Optimize

Consider $\min_{\mathbf{x}\in\mathbb{R}^n} F(\mathbf{x})$.

Classic Optimization: Designing iterative update rules $\mathbf{x}_{k+1} = \mathcal{T}_F(\mathbf{x}_k)$

Learning to Optimize (L2O): Learn an update rule from data $\mathbf{x}_{k+1} = \mathcal{T}_F(\mathbf{x}_k; \theta)$

For example,

A classic optimization algorithm: gradient descent:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla F(\mathbf{x}_k), \quad k = 0, 1, 2, \dots$$

Learn an update rule that is parameterized by neural networks[1]:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathrm{NeuralNetwork}(\mathbf{x}_k; \phi)$$

The parameters ϕ are trained via $\min_\phi \mathbb{E}_{F\in\mathcal{F}} \sum_{k=1}^{K} F(\mathbf{x}_k)$,
where $\mathcal{F}$ denotes the training set of optimization problem instances.

Such learned rules can generalize to problems similar to the training samples.

[1] Andrychowicz et al. [2016], Li and Malik [2016]

Discussions on L2O

Observation: The learned update rule may diverge on unseen instances.

How to alleviate such an issue? In the literature, some efforts have been made[2]:

- Regularizing the output of neural networks
- Improving training techniques

We consider this problem from another perspective:

- Neural networks are universal approximators.
- We actually search the update rule from such an operator space:

$$\{\mathbf{d} : \mathbb{R}^n \to \mathbb{R}^n, \mathbf{d} \text{ is continuous}\}$$

The searching space is too large!
Some operators are turely not what we want:

- No fixed point: $\mathbf{d}(\mathbf{x}) = \mathbf{x} + \mathbf{1}$
- Unable to converge: $\mathbf{d}(\mathbf{x}) = 2\mathbf{x}$

Can we explicitly remove these invalid operators from the searching space?

[2] [Wichrowska et al., 2017, Wu et al., 2018, Metz et al., 2019, Chen et al., 2020, Harrison et al., 2022, Metz et al., 2022]

A Preliminary Result

We make assumptions on the update rule and derive a rule with structure.

Core assumptions: For any sequence $\{\mathbf{x}_k\}$ generated by the given update rule

- If $\mathbf{x}_k$ is an optimal solution, then it holds that $\mathbf{x}_{k+1} = \mathbf{x}_k$
- The sequence $\{\mathbf{x}_k\}$ must converge to one of the optimal solutions.

Theorem (Informal)

For any convex and smooth f and any update rule that satisfies the above assumptions, there exist $\mathbf{P}_k \in \mathbb{R}^{n\times n}$ and $\mathbf{b}_k \in \mathbb{R}^n$ satisfying

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

with $\mathbf{P}_k$ is bounded and $\mathbf{b}_k \to \mathbf{0}$ as $k \to \infty$.

A "good" update rule is not totally free!

Instead of learning $\mathbf{d}_k$, one may learn a *preconditioner* $\mathbf{P}_k$ and a *bias* $\mathbf{b}_k$.

More results

We extend such a result to

- Convex non-smooth functions
- Update rules that take in a longer horizon

We propose a novel L2O model inspired by these theoretical results.

The proposed model has strong generalization ability.

Comparison: In-Distribution Test

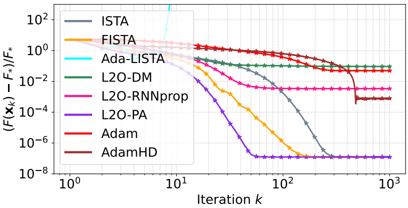

Figure: LASSO: Train and test on synthetic data.

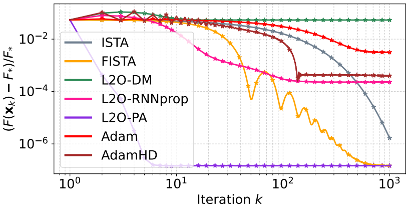

Figure: Logistic: Train and test on synthetic data.

Comparison: Out-of-Distribution Test

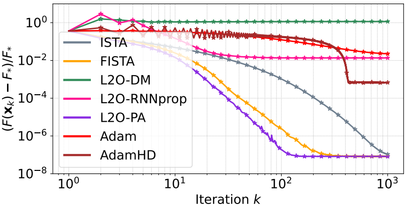

Figure: LASSO: Train on synthetic data and test on real data (BSDS500).

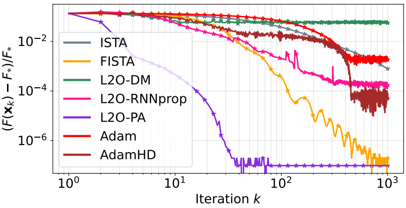

Figure: Logistic: Train on synthetic data and test on real data (Ionosphere).

Thanks for listening!

Our paper: https://openreview.net/forum?id=Tm7NpcjSE4

Our codes: https://github.com/xhchrn/MS4L2O

References:

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.

Tianlong Chen, Weiyi Zhang, Zhou Jingyang, Shiyu Chang, Sijia Liu, Lisa Amini, and Zhangyang Wang. Training stronger baselines for learning to optimize. *Advances in Neural Information Processing Systems*, 33:7332–7343, 2020.

James Harrison, Luke Metz, and Jascha Sohl-Dickstein. A closer look at learned optimization: Stability, robustness, and inductive biases. *arXiv preprint arXiv:2209.11208*, 2022.

Ke Li and Jitendra Malik. Learning to optimize. In *International Conference on Learning Representations*, 2016.

Luke Metz, Niru Maheswaranathan, Jeremy Nixon, Daniel Freeman, and Jascha Sohl-Dickstein. Understanding and correcting pathologies in the training of learned optimizers. In *International Conference on Machine Learning*, pages 4556–4565. PMLR, 2019.

Luke Metz, C Daniel Freeman, James Harrison, Niru Maheswaranathan, and Jascha Sohl-Dickstein. Practical tradeoffs between memory, compute, and performance in learned optimizers. In *Conference on Lifelong Learning Agents*, pages 142–164. PMLR, 2022.

Olga Wichrowska, Niru Maheswaranathan, Matthew W Hoffman, Sergio Gomez Colmenarejo, Misha Denil, Nando Freitas, and Jascha Sohl-Dickstein. Learned optimizers that scale and generalize. In *International Conference on Machine Learning*, pages 3751–3760. PMLR, 2017.

Yuhuai Wu, Mengye Ren, Renjie Liao, and Roger Grosse. Understanding short-horizon bias in stochastic meta-optimization. *arXiv preprint arXiv:1803.02021*, 2018.