STABILIZING BI-LEVEL HYPERPARAMETER OPTIMIZATION USING MOREAU-YOSIDA REGULARIZATION

Sauptik Dhar, Unmesh Kurup, Mohak Shah
America Research Lab, LG Electronics, Santa Clara, CA 95054

LG

Introduction

❑ **Focus**: Improving stability of Bi-level Hyperparameter Optimization for ill conditioned problems.

Hyperparameter Optimization (HPO)

❑ HPO is indispensable for optimal ML model building.

❑ Broadly two generic approaches,

– Algorithm independent approaches: Grid Search, Random Search, Bayesian Optimization, Bandit-Based Search.

– Direct Optimization approaches: Bi-Level optimization, Analytic Bound based model selection.

Bi-Level Hyperparameter Optimization

❑ Casts HPO as bi-level optimization,

$$\boldsymbol{\lambda}^* \in \underset{\boldsymbol{\lambda}}{\operatorname{argmin}}\ L_V(\boldsymbol{\lambda}, \underset{\mathbf{w}}{\operatorname{argmin}}\ L_T(\mathbf{w}, \boldsymbol{\lambda})) \qquad (1)$$

$L_V =$ Validation Loss $\quad \boldsymbol{\lambda} =$ Hyper parameters

$L_T =$ Training Loss $\quad \mathbf{w} =$ Model Parameters

❑ Popular approach SHO [1] use **best-response function**,

$$\boldsymbol{\lambda}^* \in \underset{\boldsymbol{\lambda}}{\operatorname{argmin}}\ L_V(\boldsymbol{\lambda}, G_{\boldsymbol{\phi}}(\boldsymbol{\lambda})) \text{ s.t. } G_{\boldsymbol{\phi}}(\boldsymbol{\lambda}) \in \underset{\mathbf{w}}{\operatorname{argmin}}\ L_T(\mathbf{w}, \boldsymbol{\lambda}) \qquad (2)$$

where, $G_{\boldsymbol{\phi}}(\lambda) = \lambda\phi_1 + \phi_0\ ;\ \boldsymbol{\phi} \in \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ L_T(\Lambda\ \boldsymbol{\theta}, \lambda)$

❑ Solved through alternating gradients (see SHO [1])

❑ Suffer from instabilities for ill-conditioned problems.

Contributions

❑ Propose to stabilize alternating gradient based bi-level HPOs through our Moreau-Yosida regularized algorithm.

❑ Provide convergence analysis for MY-HPO algorithm.

❑ Provide empirical results in support of our method.

Moreau-Yosida Regularized bi-level HPO (MYHPO)

❑ Reformulate the HPO problem to solve,

$$\min_{\lambda, \mathbf{w}}\quad L_T(\mathbf{w}, \lambda^*)\ +\ L_V(\lambda, G_{\boldsymbol{\phi}^*}(\lambda)) \qquad (3)$$

where, $\mathbf{w} = G_{\boldsymbol{\phi}^*}(\lambda) = \lambda\phi_1^* + \phi_0^* = \Lambda\ \phi^*, \qquad \Lambda = [\lambda\mathbf{I}|\mathbf{I}]$

$\lambda^*, \phi^* = \begin{bmatrix} \phi_1^* \\ \phi_0^* \end{bmatrix}$ - is an optimal solution given by oracle.

❑ **Proposition 1** shows solving (3) $\Rightarrow$ (2)

❑ **(MY-HPO Algorithm)** Solving (3) involves the steps,

[Step 1]: $\phi_0^{k+1} = \overline{\mathbf{v}^{k+1}} = \sum v_j^{k+1}; \quad \phi_1^{k+1} = \dfrac{\mathbf{v}^{k+1} - \overline{\mathbf{v}^{k+1}}}{\lambda^k}$

where $\mathbf{v}^{k+1} = \underset{\mathbf{v}}{\operatorname{argmin}}\ L_T^j(\mathbf{v}, \lambda^k)$

[Step 2]: $\mathbf{w}^{k+1} = \underset{\mathbf{w}}{\operatorname{argmin}}\ L_T(\mathbf{w}, \lambda^k) + (\mathbf{u}^k)^T(\mathbf{w} - \Lambda^k\phi^{k+1}) + \dfrac{\rho}{2}||\mathbf{w} - \Lambda^k\phi^{k+1}||_2^2$

[Step 3]: $\lambda^{k+1} = \underset{\lambda}{\operatorname{argmin}}\ L_V(\lambda, G_{\phi^{k+1}}(\lambda)) + (\mathbf{u}^k)^T(\mathbf{w}^{k+1} - \Lambda\phi^{k+1}) + \dfrac{\rho}{2}||\mathbf{w}^{k+1} - \Lambda\phi^{k+1}||_2^2$

[Step 4]: $\mathbf{u}^{k+1} = \mathbf{u}^k + \rho(\mathbf{w}^{k+1} - \Lambda^{k+1}\phi^{k+1})$

❑ Theoretical convergence analysis are provided in **Prop. 2** and **Claim 1**.

❑ We take gradient updates for Steps 2 and 3 (for Results).

Intuition

❑ MY regularization of f is $f_{1/\rho}(\cdot) := \min_x\ f(x) + \frac{\rho}{2}||\cdot - x||_2$

❑ Steps 2, 3 minimize Moreau-Yosida regularized L_T, L_V which adds stability under ill-conditioned settings.

❑ Step 4 lends to additional stability by ensuring the primal constraint $\mathbf{w} = \Lambda\ \phi$ is not considerably violated.

Results

❑ German Traffic Sign '30' vs. '80' Recognition Data:

– Validation Loss: $L_V = \frac{1}{N_V}\sum_{\mathbf{x}_i \in \mathcal{V}} \log(1 + e^{-y_i\mathbf{w}^T\mathbf{x}_i})$

– Training Loss: $L_T = \frac{1}{N_T}\sum_{\mathbf{x}_i \in \mathcal{T}} \log(1 + e^{-y_i\mathbf{w}^T\mathbf{x}_i}) + e^{\lambda}||\mathbf{w}||_2^2$

– Train (N_T) ,Val (N_V) , Test set size = 1000. Dimension = 1568 (HOG)

Table 1: Loss values compared to popular HPO algorithms in Auptimizer [2].

HPO	SHO	MyHPO (C)	MyHPO (BT)	Random	Grid	HyperOpt	Spearmint
Train ($\times 10^{-2}$)	6.83 ± 1.04	3.26 ± 2.5	$\mathbf{4.31 \pm 1.2}$	47.48 ± 117.3	0.09 ± 0.01	11.9 ± 30.5	0.09 ± 0.01
Val. ($\times 10^{-2}$)	14.35 ± 2.34	14.35 ± 2.38	$\mathbf{13.98 \pm 2.3}$	39.85 ± 75.4	23.5 ± 6.6	18.9 ± 11.7	23.5 ± 6.6
Test ($\times 10^{-2}$)	14.04 ± 1.82	13.8 ± 2.25	$\mathbf{13.59 \pm 1.9}$	38.6 ± 70.6	21.8 ± 5.2	17.6 ± 8.5	21.8 ± 5.2

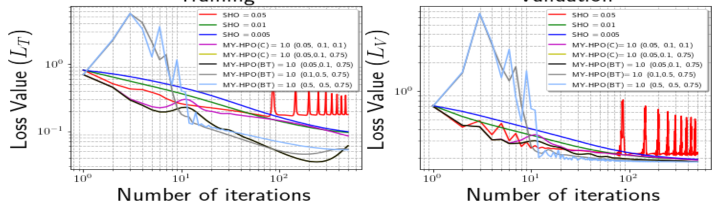

Fig.1 Convergence behavior of SHO vs. MY-HPO for different step-sizes.

– MY-HPO algorithm outperforms the baseline algorithms.

– SHO destabilizes for higher step sizes (Fig. 1)

– MY-HPO accommodates higher step-size and improves convergence.

– Additional results available in paper.

Summary

❑ Proposed MY-HPO algorithm to stabilize bi-level HPO.

❑ Provide convergence guarantees for MY-HPO algorithm.

❑ Provide empirical results in support of our method.

REFERENCES
1. "Stochastic hyperparameter optimization through hypernetworks" arXiv:1802.09419,
2. Auptimizer an extensible, open-source framework for hyperparameter tuning. arXiv:1911.02522

Email : {sauptik.dhar, unmesh.kurup, mohak.shah}@lge.com

7th ICML Workshop on Automated Machine Learning (2020)