Stability and Generalization of (Gradient-Based) Bilevel Programming in Hyperparameter Optimization

Chongxuan Li

Gaoling School of AI, Renmin University of China

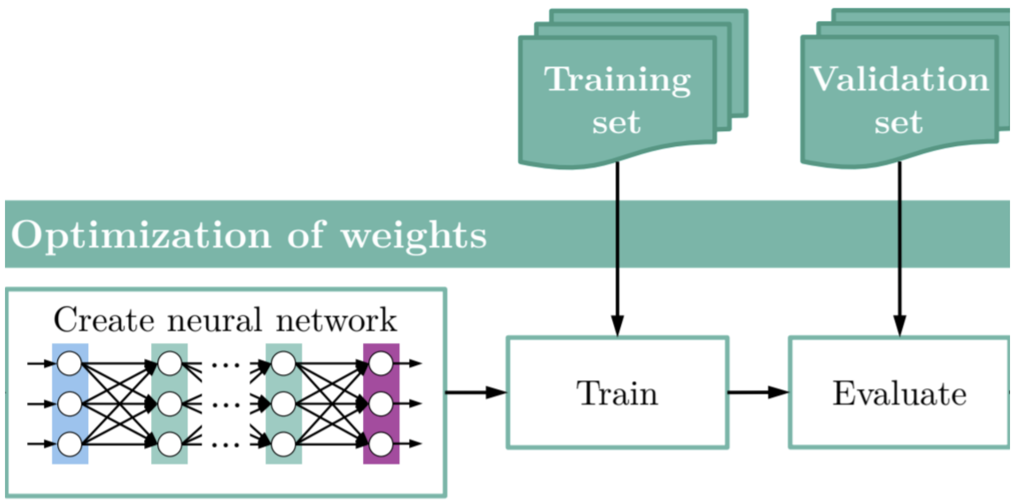
Fan Bao Guoqiang Wu Jun Zhu Bo Zhang

Outline

- Background on hyperparameter optimization
- Two approaches for hyperparameter optimization:
  - Search-based cross validation
  - Gradient-based bilevel programming
- Stability and generalization for gradient-based bilevel programming
- Conclusion and discussion

Background on hyperparameter optimization

Learning paradigm



Informally, a *learning algorithm* optimizes the *model parameters* according to a *certain loss* on a sample called *training set*, and hopefully, the optimized model will perform well on *unseen data of the same task*.

Statistical learning theory

- Goal: minimize *expected risk* with respect to a target distribution

- Algorithm: minimize *empirical risk* on a sample following the target distribution
  - The examples in the sample are independent and identically distributed

- Empirical risk is an *(unbiased) Monte Carlo estimate* of expected risk
  - Gap is caused by the *randomness* of the finite-size sample

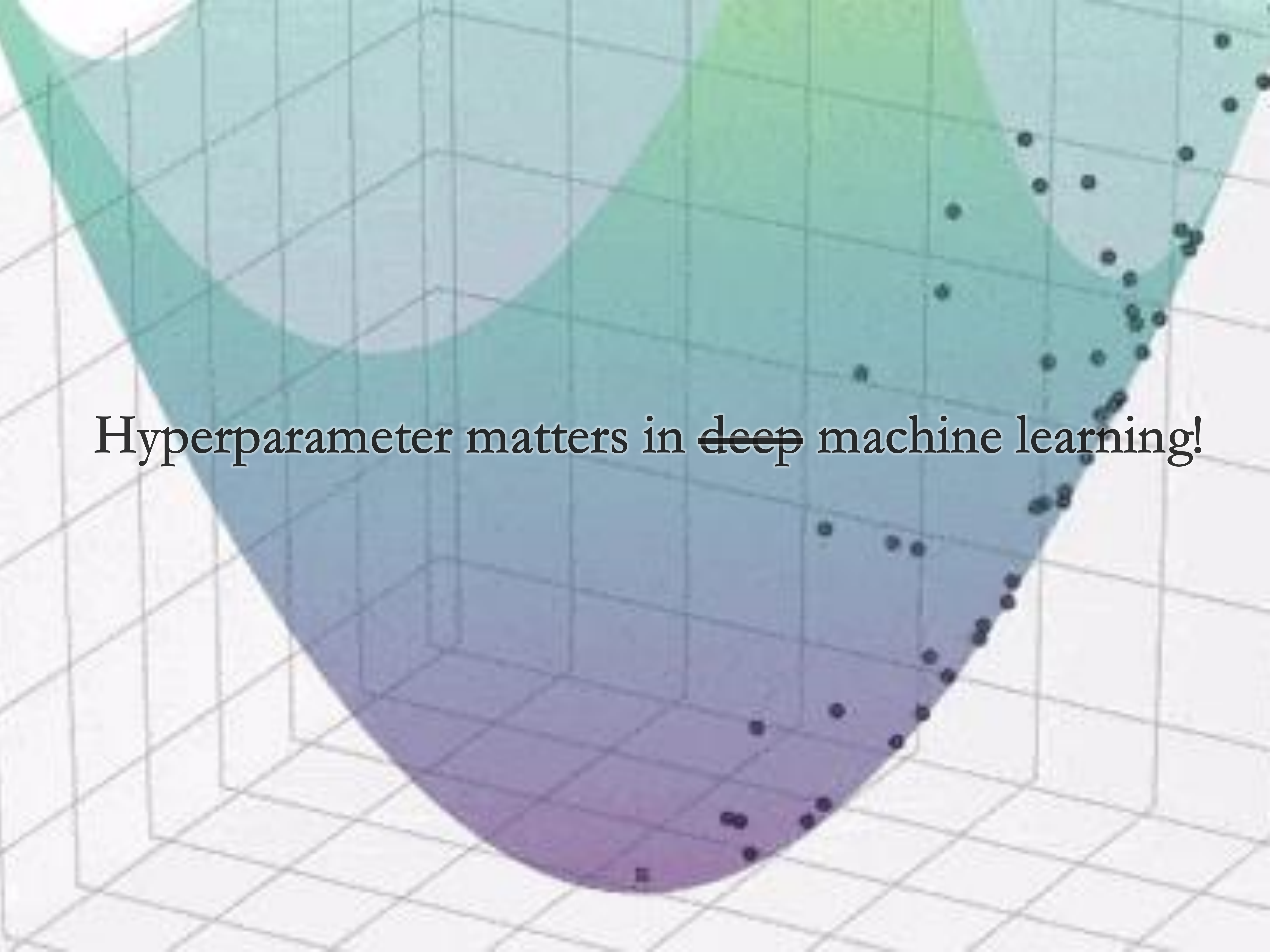
- Theoretical guarantee by *concentration inequalities*
  - With a high probability over the draw of the sample, the gap is small

Hyperparameters

- Model parameters: optimized by the learning algorithm on the training set

- *Hyperparameters*: specified as inputs to the learning algorithm
  - *Model selection*
    - Weight of regularizations, number of parameters, topology of neural networks
  - Others
    - Learning rate, mini-batch size, data coefficients, parameter initialization

- Hyperparameters are selected (or optimized) on a *validation set*



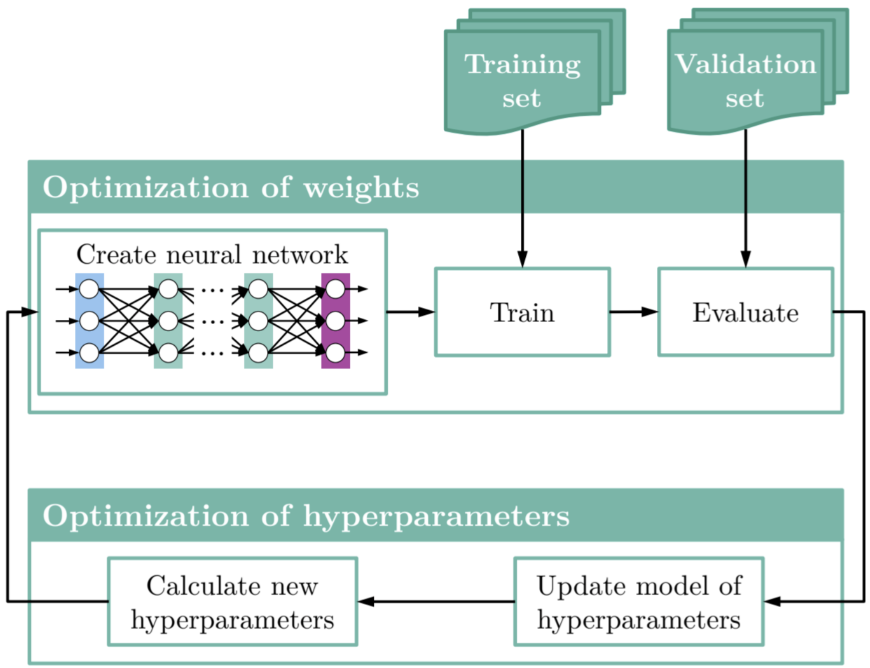
Hyperparameter matters in ~~deep~~ machine learning!

Learning paradigm with hyperparameter optimization

Outer level: seek the best hyperparameter on validation data

Inner level: seek the best parameter on training data given hyperparameter

Current hyperparameters

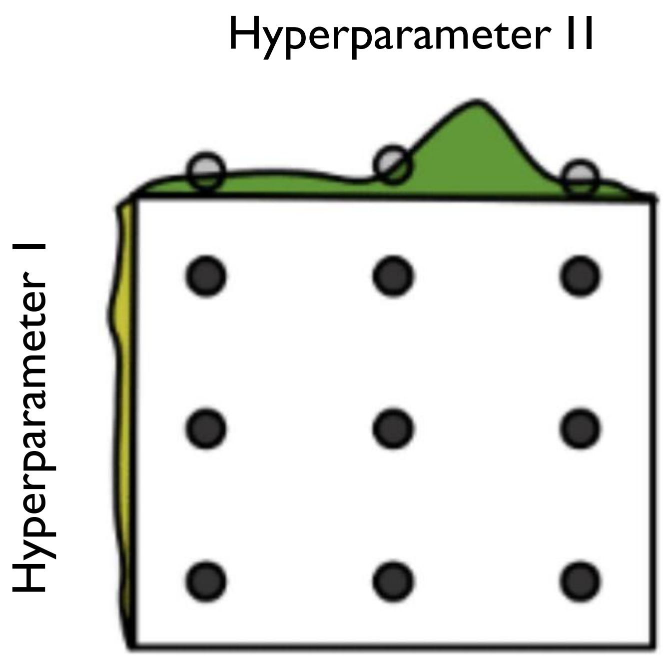


Trained parameters given current hyperparameters

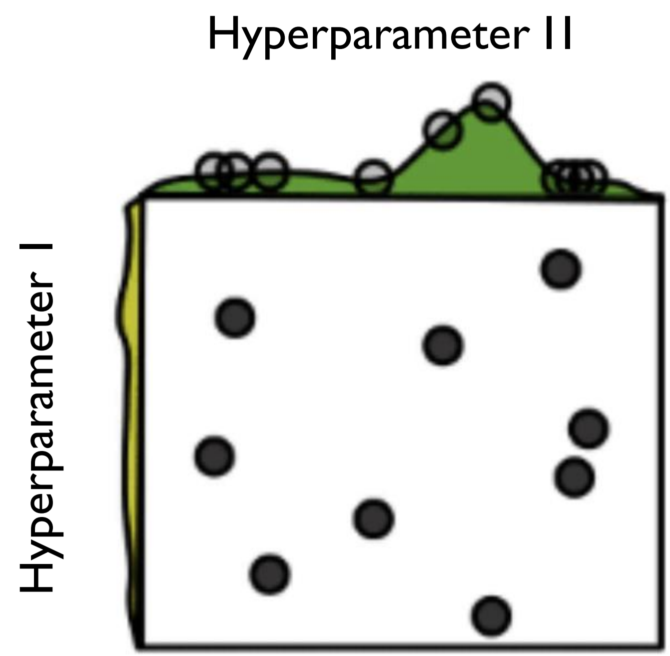
Two approaches for hyperparameter optimization

Classical search-based cross validation

A two-dim example



Grid search



Random search

Classical search-based cross validation

Get T prefixed hyperparameters by grad search or random search

Outer level:

Inner level: for each hyperparameter

Get the corresponding parameters by (approximate) empirical

risk minimization (e.g. using SGD) on training set

Select the best parameter-hyperparameter pair on validation set

Existing theory: a high probability bound of expected risk based on empirical risk on validation set.

Search-based cross validation in practice

- Search based CV is simple and widely used in research and industry
  - Number of hyperparameters is around $10^0 \sim 10^2$

- *Curse of dimensionality*
  - The search does not leverage the validation data and thus inefficient
  - The search space grows exponentially with respect to the number of hyperparameters

- Explicitly use the information of validation data during search to scale up

Gradient-based bilevel programming

Initialize parameters and hyperparameters

Outer level:

Inner level: given the current hyperparameter

Update parameters by GD or SGD of K steps on training set

Update hyperparameters by GD or SGD of 1 step on validation set

(trained parameters are functions of hyperparameters)

Output the hyperparameter-parameter pair after T steps of outer level

Gradient-based bilevel programming

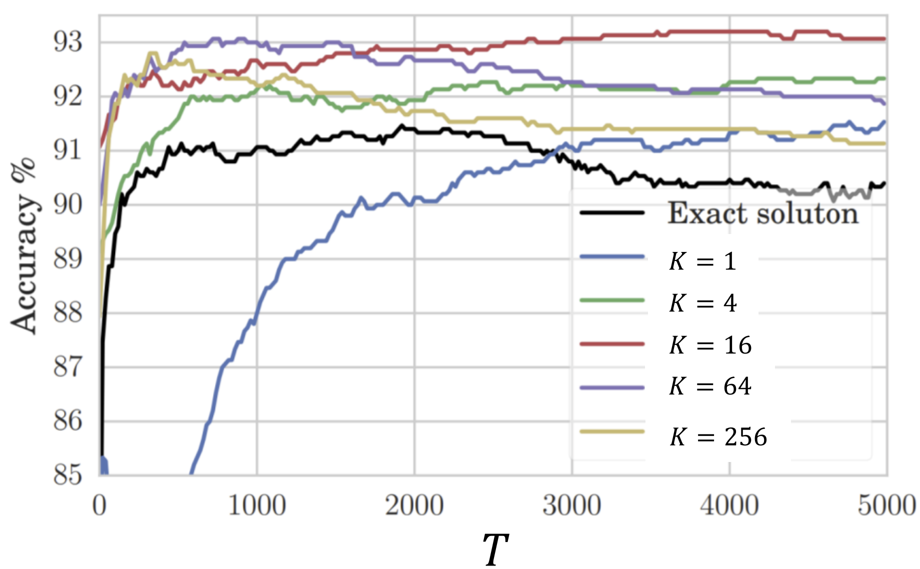
- It is referred to as unrolled differentiation (UD)
  - UD exploits the gradient information on validation data during search
  - For scalability and efficiency, SGD is preferable
  - Large scale HO: number of hyperparameters $10^4 \sim 10^6$

- Examples of UD
  - Differentiable neural architecture search [1]
  - Feature learning [2]
  - Data reweighting for imbalanced or noisy samples [3]

Theory of UD

Franceschi et al., Bilevel Programming for Hyperparameter Optimization and Meta-Learning, ICML 2018.

- Optimization: existence of optimal solution and convergence are proved
- Generalization: no rigorous analysis and there exist mysterious behaviors



For a large K and T, the algorithm may overfit to the validation set:
test accuracy decreases when optimizing the validation loss.

Motivation

- This talk takes a first step towards analyzing UD in the perspective of statistical learning theory and answering the following questions rigorously:
  - *Can we obtain certain learning guarantees for UD and explain its practical behavior?*
  - *When should we prefer UD over classical CV approaches in a theoretical perspective?*
  - *Can we develop new algorithms that improve UD with theoretical guarantee?*

Stability and generalization for gradient-based bilevel programming

Settings and notations

- Data space Z, parameter space Θ, hyperparameter space Λ
- Target distribution D, S is a i.i.d. sample drawn from D
  - Two samples: S^{tr} of size n and S^{val} of size m
- Loss function $L: \Theta \times \Lambda \times Z \to [a, b]$
  - $L_z(\lambda, \theta)$ denotes the value of L evaluated on example z given λ, θ
- HO algorithm $A: Z^n \times Z^m \to \Theta \times \Lambda$, the output is denoted as $A(S^{\text{tr}}, S^{\text{val}})$
  - Randomized HO algorithm outputs a random variable on $\Theta \times \Lambda$

Risks and generalization gap

- *Expected risk* of $(\lambda, \theta) \in \Lambda \times \Theta$ with respect to a target distribution D

$$R(\lambda, \theta) = \mathop{\mathrm{E}}_{z \sim D}[L_z(\lambda, \theta)]$$

- *Empirical risk* of $(\lambda, \theta) \in \Lambda \times \Theta$ on validation set $S^{\mathrm{val}} = \{z_1, \ldots, z_m\}$

$$\hat{R}^{\mathrm{val}}(\lambda, \theta) = \frac{1}{m} \sum_{i=1}^{m} L_{z_i}(\lambda, \theta)$$

- Our goal is to bound the *generalization gap*

$$|R(\lambda^{UD}, \theta^{UD}) - \hat{R}^{\mathrm{val}}(\lambda^{UD}, \theta^{UD})|$$

Gradient-based bilevel programming

Initialize λ_0 and θ_0

Outer level: for $t = 0, \ldots, T-1$

Inner level: for $k = 0, \ldots, K-1$

$$\theta_{k+1}^{t} \leftarrow \theta_k^t - \eta_k^t \times \nabla_\theta \hat{R}^{\mathrm{tr}}\left(\lambda_t, \theta, S^{\mathrm{tr}}\right)|_{\theta=\theta_k^t} \text{ (SGD)}$$

$$\lambda_{t+1} \leftarrow \lambda_t - \alpha_t \times \nabla_\lambda \hat{R}^{\mathrm{val}}\left(\lambda, \theta_K^t(\lambda), S^{\mathrm{val}}\right)|_{\lambda=\lambda_t} \text{ (SGD)}$$

Output $A^{UD}\left(S^{\mathrm{tr}}, S^{\mathrm{val}}\right) \leftarrow (\lambda_T, \theta_K^T)$

Notion of stability

Bao et al. Stability and Generalization of Bilevel Programming in Hyperparameter Optimization, NeurIPS 2021.

Definition 1: A randomized HO algorithm $\boldsymbol{A}$ **is** $\boldsymbol{\beta}$**-uniformly stable on validation in expectation** if for all validation datasets $S^{\text{val}}, S'^{\text{val}} \in Z^m$ such that $S^{\text{val}}, S'^{\text{val}}$ differ in at most one sample, we have

$$\forall S^{\text{tr}} \in Z^n, \forall z \in Z, \mathop{\mathrm{E}}_{A}\left[L_z\left(A\left(S^{\text{tr}}, S^{\text{val}}\right)\right) - L_z\left(A\left(S^{\text{tr}}, S'^{\text{val}}\right)\right)\right] \leq \beta.$$

- Stable: a small perturbation in data won't change the loss of the algorithm too much (risks are losses averaged by different data)
- Uniform: there exists a stability coefficient for all configurations
- On valuation: HO seeks the best pair based on validation performance
- In expectation: randomness in algorithm (e.g. SGD)

Stability based generalization bound

Bao et al. Stability and Generalization of Bilevel Programming in Hyperparameter Optimization, NeurIPS 2021.

Theorem 1 (Generalization bound of a uniformly stable algorithm)
Suppose a randomized HO algorithm A is β-uniformly stable on validation in expectation , then

$$\left|\underset{A,S^{\mathrm{tr}},S^{\mathrm{val}}}{\mathrm{E}}\left[R\left(A\left(S^{\mathrm{tr}},S^{\mathrm{val}}\right)\right)-\hat{R}^{\mathrm{val}}\left(A\left(S^{\mathrm{tr}},S'^{\,\mathrm{val}}\right)\right)\right]\right| \leq \beta.$$

- The proof follows Definition 1 and convexity of $f(x) = |x|$.
- The theorem holds for any stable algorithm, not restricted to UD
- It is algorithm dependent, not a uniform bound (e.g., complexity based)
- It is an expectation bound, not a high probability bound.
    - Here a high probability bound is nontrivial because of A is random

Stability of UD

Bao et al. Stability and Generalization of Bilevel Programming in Hyperparameter Optimization, NeurIPS 2021.

Theorem 2 (Stability of UD with SGD, informal) Suppose the outer level optimization (depending on $\hat{\theta}(\lambda)$) is L-Lipschitz continuous and γ-Lipschitz smooth w.r.t. λ. Then, UD with SGD (T steps and a sufficiently small learning rate) is β-uniformly stable on validation in expectation with $\beta = \tilde{O}(\frac{1}{m} L^{\frac{1}{1+\gamma}} T^{\frac{\gamma}{1+\gamma}})$, which is increasing w.r.t. L and γ.

- Proof follows the stability analysis of SGD in nonconvex optimization [5]
- Consider outer level steps and size of validation set, $\beta = \tilde{O}\left(\frac{T^{\kappa}}{m}\right), \kappa < 1$
- Requirement of the inner level optimization
  - It should result in a smooth outer level optimization problem
  - It does not necessarily be gradient-based

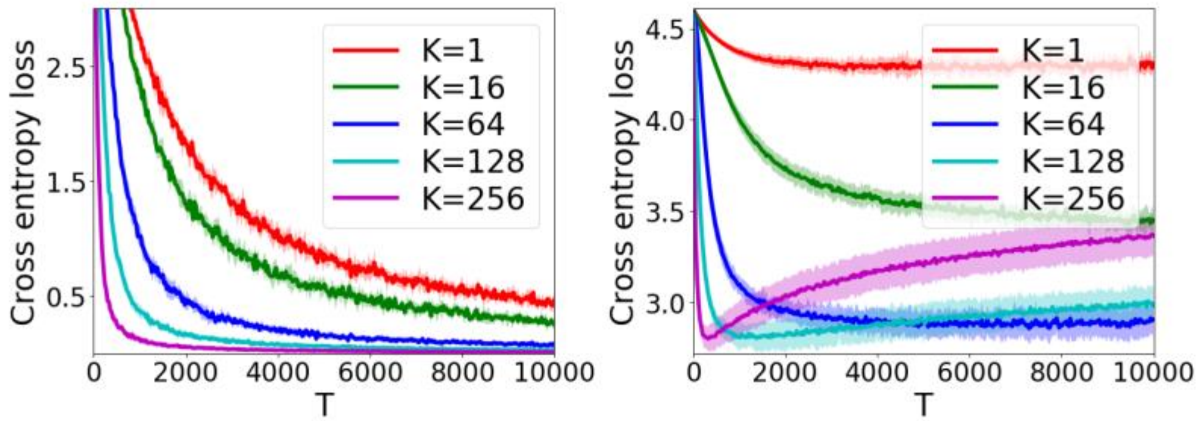
Smoothness of inner level optimization

- **Theorem 3 (Smoothness of UD, informal)** Under smoothness assumptions on model and loss, if the inner level problem is solved with K steps SGD with learning rate η, then the outer level optimization as a function of λ is Lipschitz continuous with $L = \tilde{O}((1+\eta)^K)$ and Lipschitz smooth with $\gamma = \tilde{O}((1+\eta)^{2K})$.

- Proof follows the smoothness assumptions on model and loss function
- In terms of inner level steps, $\beta = \tilde{O}((1+\eta)^{2K}/m)$
  - In nonconvex optimization (e.g., using DNNs), it is tight w.r.t. K
- Improved results under stronger assumptions
  - If the inner level optimization is convex, $\beta = O(K^2)$
  - If the inner level optimization is strongly convex, $\beta = O(1)$

Implication: explaining the practical behavior of UD

UD may overfit to validation set given a large K. $\beta = \tilde{O}((1+\eta)^{2K})$



(a) Validation loss (FL) (b) Testing loss (FL)

The results agree with existing work and support our theory

Regularized UD is more stable

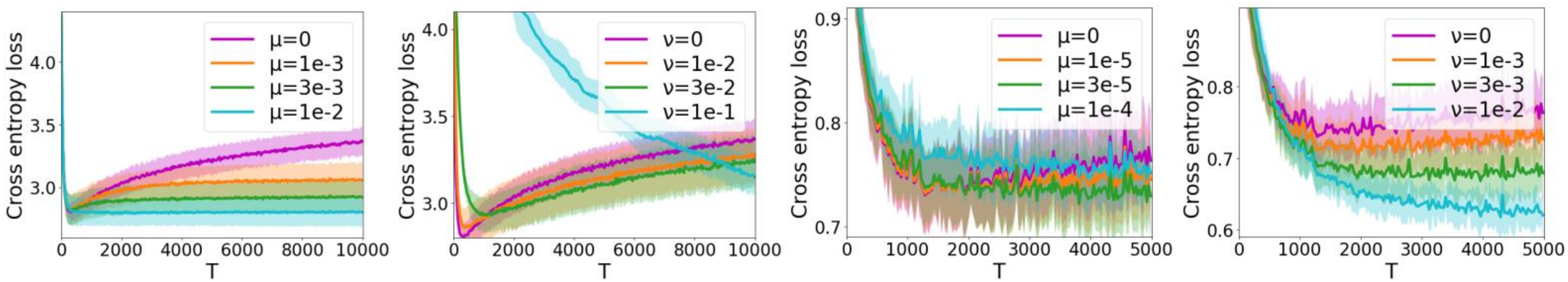
Outer level: $\lambda^* = \arg\min_{\lambda \in \Lambda} \hat{R}^{\mathrm{val}}\left(\lambda, \theta^*(\lambda), S^{val}\right) + \frac{\mu}{2} ||\lambda||_2^2$

Inner level: $\theta^*(\lambda) = \arg\min_{\theta \in \Theta} \hat{R}^{\mathrm{tr}}\left(\lambda, \theta, S^{\mathrm{tr}}\right) + \frac{\nu}{2} ||\theta||_2^2$

- **Proposition1 (Stability of regularized UD, informal)** Under the same assumptions in Theorem 2 and Theorem 3, if μ and ν are sufficiently small, then regularized UD has a smaller stability coefficient.
  - $\beta = \tilde{O}(T^\kappa/m)$ with a smaller κ related to μ
  - $\beta = \tilde{O}((1 + c\eta)^{2K}/m)$ with a smaller c related to ν

Regularized UD is more stable

Regularizations of both λ and θ can relieve overfitting



(a) Reg-outer (FL) (b) Reg-inner (FL) (c) Reg-outer (DR) (d) Reg-inner (DR)

There is no clear winner of Reg-outer (λ) and Reg-inner (θ)

Search-based cross validation

Get $\lambda_1, \dots \lambda_T$ by grad search or random search

For $k = 1, \dots, K$

$$\theta_{k+1}^t \leftarrow \theta_k^t - \eta_k^t \times \nabla_\theta \hat{R}^{\mathrm{tr}}(\lambda_t, \theta, S^{\mathrm{tr}})|_{\theta=\theta_k^t} \text{ (SGD)}$$

Select the t^* on validation set and $A^{CV}(S^{\mathrm{tr}}, S^{\mathrm{val}}) \leftarrow (\lambda_{t^*}, \theta_K^{t^*})$

- A classical result: with a probability at least $1-\delta$, the following holds

$$\forall\, t \in [1, T], \qquad R(\lambda_t, \theta_K^t) \le \hat{R}^{\mathrm{val}}(\lambda_t, \theta_K^t) + \sqrt{\frac{\log T + \log 2/\delta}{2m}}$$

Including t^*, namely $A^{CV}(S^{\mathrm{tr}}, S^{\mathrm{val}})$

Expectation bound for classical CV

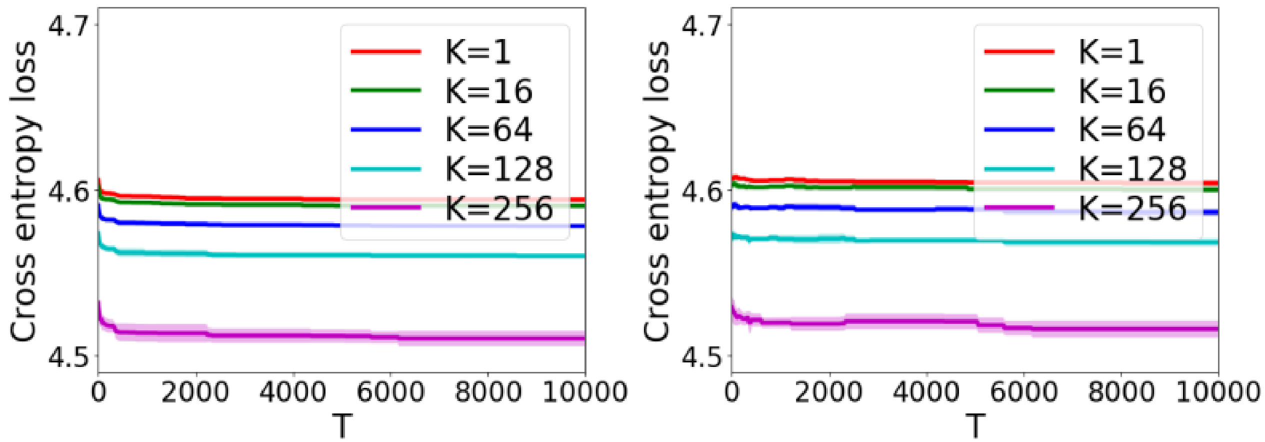
Theorem 3 (Expectation bound of CV, informal) Given any prefixed T hyperparameters, the following holds for CV algorithm

$$\left|\mathop{\mathrm{E}}_{S^{\mathrm{tr}},S^{\mathrm{val}}}\left[R\left(A^{CV}\left(S^{\mathrm{tr}},S^{\mathrm{val}}\right)\right)-\hat{R}^{\mathrm{val}}\left(A^{CV}\left(S^{\mathrm{tr}},S'^{\mathrm{\,val}}\right)\right)\right]\right| \leq O(\sqrt{\log T/m}).$$

- The expectation bound has the same order as the classical high probability bound
- Trade-off between CV and UD
  - Validation error: UD has smaller validation error
  - Generalization gap:
    - CV is $O(\sqrt{\log T})$ while UD is $\tilde{O}((1+\eta)^{2K})$ or $\tilde{O}(T^{\kappa})$
    - CV is $O(\sqrt{1/m})$ while UD is $\tilde{O}(1/m)$

Implication on CV

Cross validation with grad search hardly overfits



(a) Validation loss (FL) (b) Testing loss (FL)

In our experiments, CV is worse than UD because the validation loss is high

Conclusion and discussion

Conclusion

- We present a stability and generalization analysis of SGD-based bilevel programming in hyperparameter optimization
  - A long inner loop is unstable and can make UD overfit to validation
  - When the validation set is of relatively large size, the inner loop and outer loop are reasonably short, UD is better than CV (in expectation)
  - Regularization terms at both levels improve stability and obtain a better learning guarantee if the validation performance remains

Future work

- Tighter bounds with advanced techniques
- Comparison with bounds on training data
    - Optimal split of train-validation data
- High probability bounds
- Other algorithms
    - Implicit differentiation

中国人民大学高瓴人工智能学院
Gaoling School of Artificial Intelligence, Renmin University of China

Thanks!

Email: chongxuanli@ruc.edu.cn
Homepage: https://gsai.ruc.edu.cn/chongxuan
Office: 中国人民大学信息楼 343

Additional references

- Credit of the GIF in the first page: Karen Pleasant, https://www.pavlovsk.org/a-tutorial-on-optimizing-particle-swarm-in-python/
- Credit of the figure in page 4: Simon Claus Stock et al., A system approach for closed-loop assessment of neuro-visual function based on convolutional neural network analysis of EEG signals.
- Credit of the figure in page 10: Peter Worcester, https://medium.com/@peterworcester_29377/a-comparison-of-grid-search-and-randomized-search-using-scikit-learn-29823179bc85
- [1] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055, 2018.
- [2] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In International Conference on Machine Learning, pages 1568–1577. PMLR, 2018.
- [3] Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated backpropagation for bilevel optimization. In The 22nd International Conference on Artificial Intelligence and Statistics, pages 1723–1732. PMLR, 2019.
- [4] Mohri M, Rostamizadeh A, Talwalkar A. Foundations of machine learning[M]. MIT press, 2018.
- [5] Hardt M, Recht B, Singer Y. Train faster, generalize better: Stability of stochastic gradient descent[C]//International Conference on Machine Learning. PMLR, 2016: 1225-1234.