Learning Provably Improves the Convergence of Gradient Descent

Qingyu Song†, Wei Lin‡, Hong Xu‡

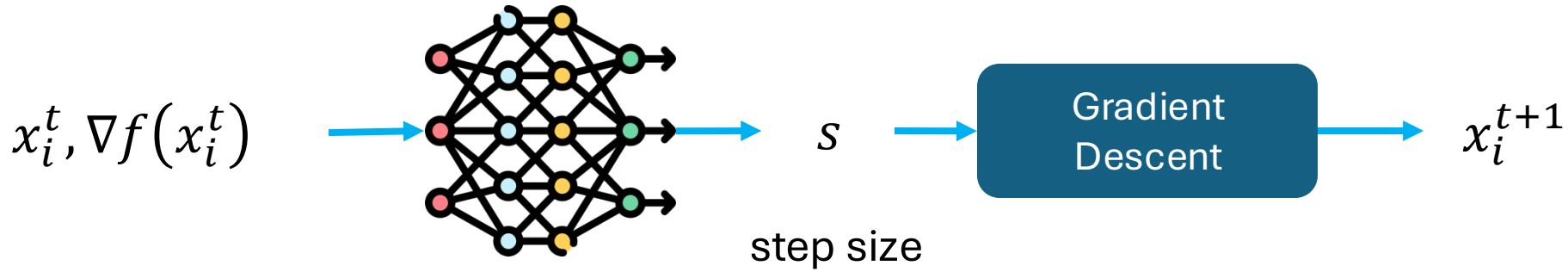
†Xiamen University, ‡The Chinese University of Hong Kong,

厦門大學
XIAMEN UNIVERSITY

香港中文大學
The Chinese University of Hong Kong

Introduction: Optimization and L2O

- Scenarios: General Optimization Problems
- Learning to Optimize (L2O) Architecture:
  - Unrolling (Math-L2O [Liu et al., ICML 2023])



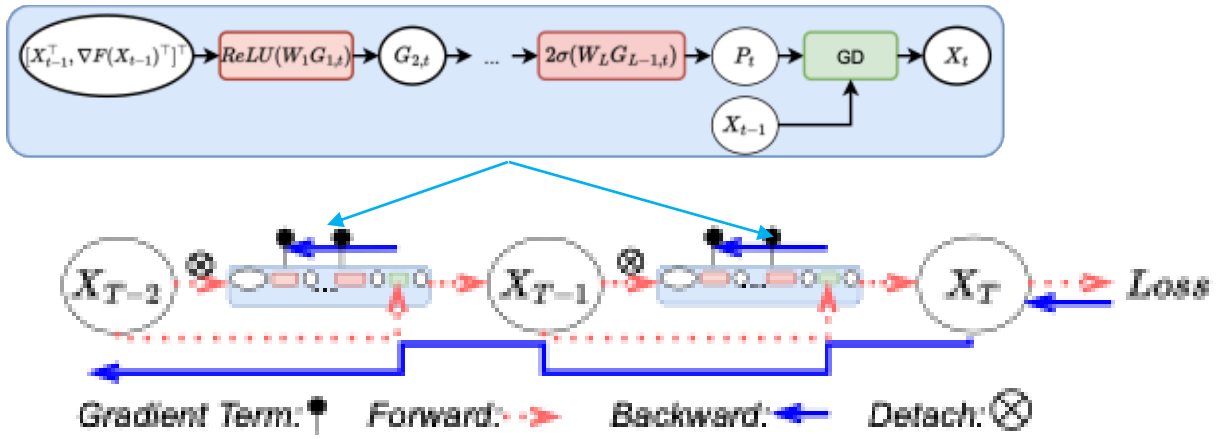
- Problem Definition: Quadratic Programming

$$\min_x f(x), x \in R^n$$

  - $f: R^n \to R$, quadratic.
  - f is convex and smooth.

Analysis: Training V.S. Unrolling

- Optimization (Unrolling) Steps v.s. Training Iterations
  - Optimization Steps (Unrolling process): $x_{t-1} \to x_t$.



  - Training Iterations: Stable T, $f(x_T^{k-1}) \to f(x_T^k)$.

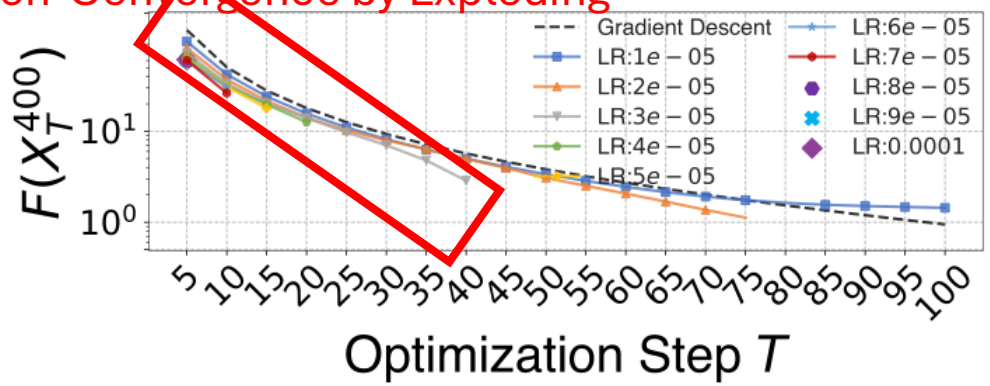
- Optimization is orthogonal to training.
  1. Can we have convergence of Unrolling-L2O from training?

Convergence with Different Unrolling Steps

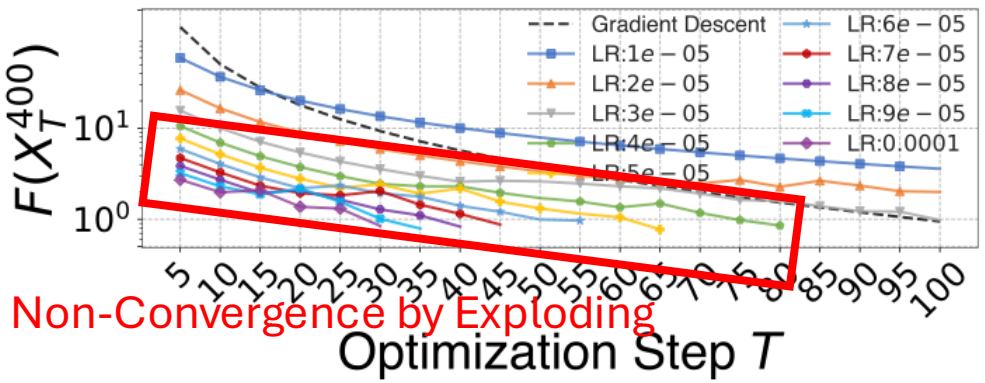
- Objective by *100* steps, by 400 training iterations

Math-L2O [Liu et al., ICML 2023]

Non-Convergence by Exploding



LISTA-CPSS [Chen et al., NeurIPS 2018]



Non-Convergence by Exploding

- Longer optimization (unrolling) steps require smaller learning rates.
- Larger learning rates improve convergence.

There is no guarantee that training will always converge.

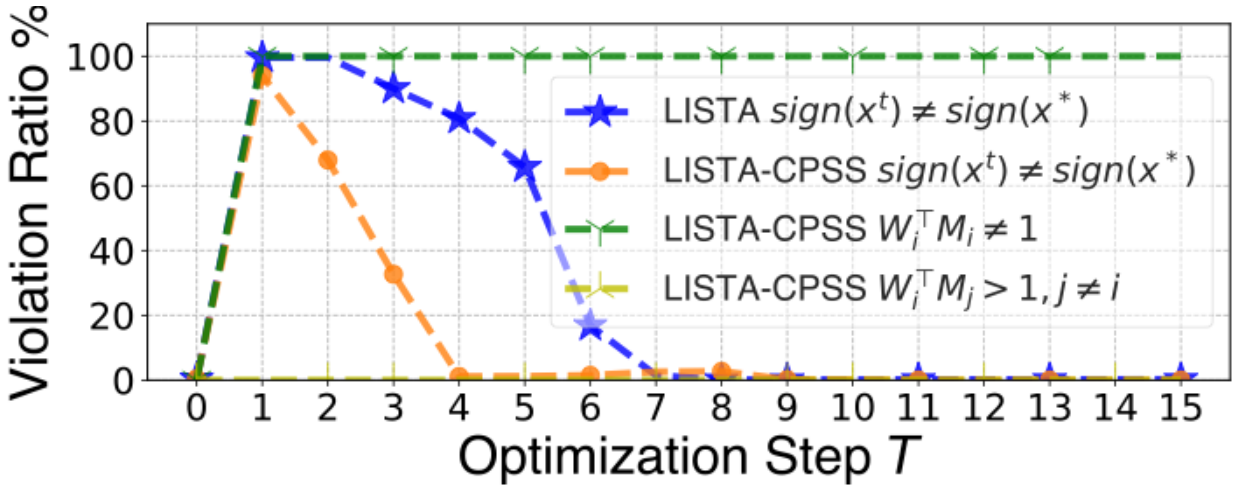
Motivate our rigorous L2O convergence proof of training.

Existing Theoretical Convergence Proofs

- Rely on Strict Assumptions
  - LISTA-CPSS [Chen et al., NeurIPS 2018]
    - Sequence converges to optimum in a fixed direction: $\text{sign}(X_t) = \text{sign}(X^*)$.
    - Learnable matrix is (semi-)orthogonal to coefficient matrix in objective:

$$\mathbf{W}_i^\top \mathbf{M}_i = 1 \text{ and } \mathbf{W}_i^\top \mathbf{M}_j > 1 \text{ for all } j \neq i.$$

- **Violated** by Real Training



Non-Trivial Unrolling-L2O Convergence Proof?

- A Novel Convergence Demonstration Methodology for Math-L2O (Unrolling L2O)
- Main Design:
  1. L2O Convergence Construction

     Convergence alignment between training and optimization.

     (NOTE: Training is orthogonal to optimization process in Unrolling L2O.)
  2. Training Convergence Demonstration.

     Neural Tangent Kernel Theorem.
  3. Deterministic Initialization Method.

     Initialize NN to achieve alignment.

     Ensure training converge.

Design Part-I: L2O Convergence Construction

- Align **Upper Bound** of L2O Training to **Lower Bound** of Gradient Descent.

1. General Convergence Rate of L2O Training:
   - k: training iteration, t: optimization step.

$$F\left(X_T^k\right) \leq r_k C\left(X_T^0\right), X_T^0 = \mathrm{L2O}_{W_{[L]}}(X_0)$$

Some Constant Mapping

2. Convergence Rate of Gradient Descent:

$$F\left(X_T^0\right) \leq \frac{\beta}{T}\left\|X_0^0 - X^*\right\|_2^2$$

3. Alignment: Let $C(X_T^0) = F(X_T^0), F\left(X_T^k\right) \leq r_k \frac{\beta}{T}\left\|X_0^0 - X^*\right\|_2^2$
4. Given constant intial point that $X_0^0 = X_0^k$, convergence rate (at step T) of L2O:

$$F\left(X_T^k\right) \leq r_k \frac{\beta}{T}\left\|X_0^k - X^*\right\|_2^2$$

Requirement: Demonstrate the convergence rate of L2O Training.

Design Part-II: Training Convergence Demonstration

- We prove it by Neural Tangent Kernel Theorem.
  - NN training: Gradient descent.
  - L-1 layer is wide: More neuron than training samples.

1. Gradient of NN Layers: By chain rule, similar to BPTT (Sec. 5.1.4, Page 86)
2. NN's outputs in L2O are upper-bounded by **inputs** (Lemma 5.3.6, Theorem 5.3.7, Page 96-97)

   Help us manipulate output of NN by the given inputs.
3. Gradient is Upper-Bounded by **Objective** (Theorem 5.3.11, page 98)

   Help us derive bounding target ($F(X_T^0)$) from NN's gradient of training.
4. L2O is A Semi-Smooth to Its Parameters (Theorem 5.3.13, page 99)

   Help us bound dynamic of sequence with dynamic of training parameters (gradient + learning rate).
5. Linear Convergence of L2O Training (Theorem 5.3.15, page 99-100)

   Conditions:
   1. *L-1 layer's singular value is **lower-bounded** by non-negative constants.*
   2. *Learning rate is **upper-bounded** by L-1 layer's singular value.*

Design Part-III: Deterministic Initialization Method

1. Hold the conditions for linear training convergence rate.
2. Achieve convergence alignment.

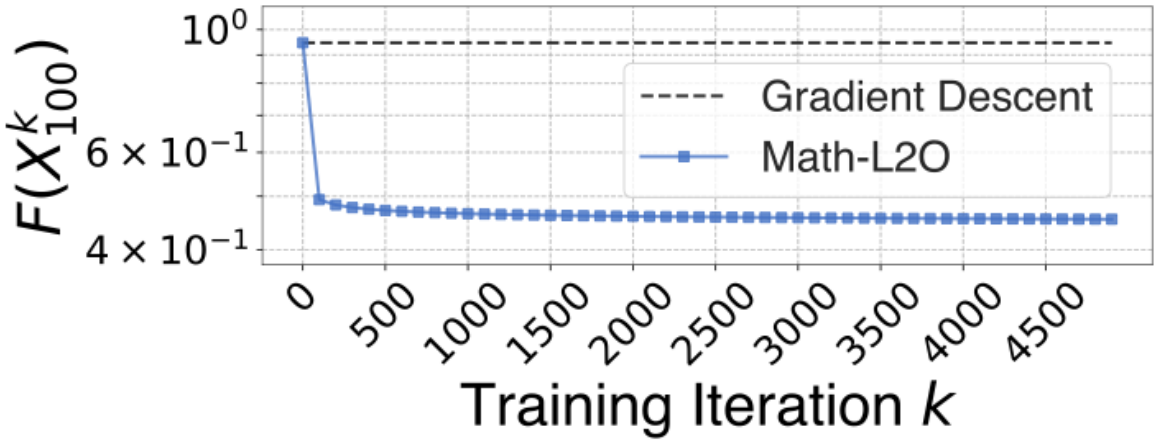
1. L2O generates step sizes of vanilla GD (alignment).

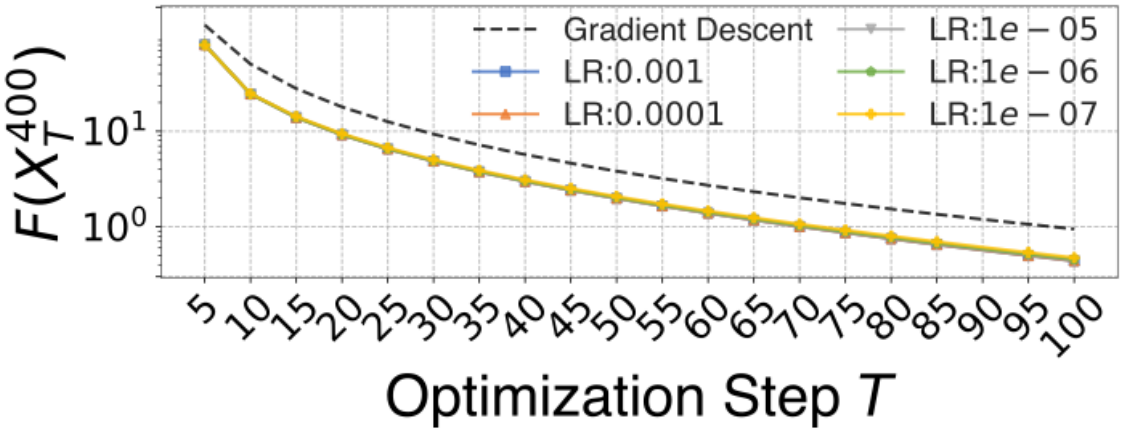
   Let parameter matrix of last layer to be $\mathbf{0}$ + 2 Sigmoid activation.

2. Positive Singular Value
   - Non-Zero: Random generated thin matrix (transposed fat).
   - Positive: QR-decomposition + absolute operation.
3. Enlarge Singular Value: Hold the Conditions of Linear Rate
   - Enlarge all entries of parameter matrices by a positive constant.
   - Proper positive constants : Lemmas 5.4.1-5.4.4 (pages 113-114).

Empirical Evaluation: Training Convergence

- Problem scale: Quadratic Programming, 512*400
- SOTA Unrolling:
  - Math-L2O [Liu et al., ICML 2023]
  - LISTA-CPSS [Chen et al., NeurIPS 2018]





- All converge
- Robust to learning rate

Empirical Evaluation: Ablation for Initialization and Learning Rate

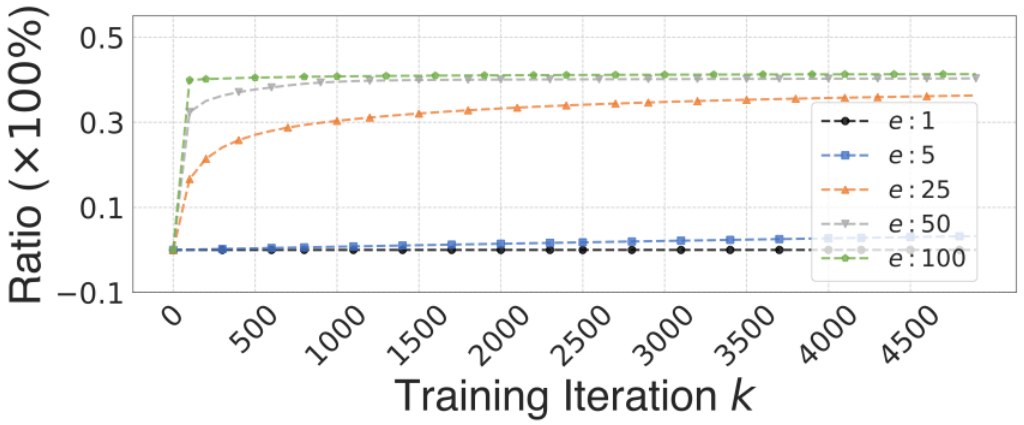
- Small Scale: QP 32*25.

Learning Rate Ablation

Borderline

- If LR is small enough, increasing LR improves convergence
- If LR is too large, convergence deteriorates.

Initialization Ablation



- Increases singular value improves convergence

NEURAL INFORMATION PROCESSING SYSTEMS

Thank You!

Project at https://github.com/NetX-lab/MathL2OProof-Official

Paper

Code

XIAMEN UNIVERSITY

The Chinese University of Hong Kong