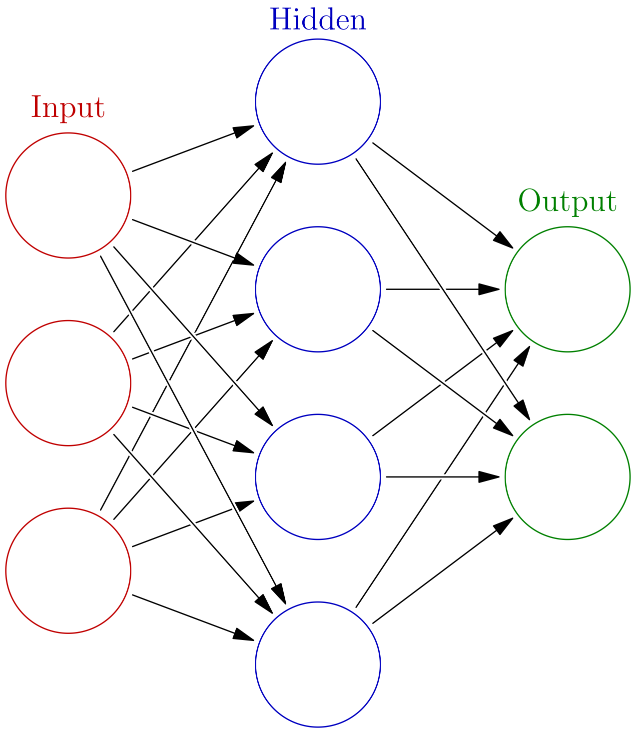
Symbolic Discovery of Optimization Algorithms

Xiangning Chen
Chen Liang
Da Huang
Esteban Real
Kaiyuan Wang
Yao Liu
Hieu Pham
Xuanyi Dong
Thang Luong
Cho-Jui
Hsieh Yifeng
Lu Quoc
V. Le

Content

Introduction

- Optimization algorithms play a crucial role in training deep neural networks.
- Deep neural networks are a cornerstone of modern artificial intelligence.
- Discovering efficient optimization algorithms for these networks is a complex task.
- The paper 'Symbolic Discovery of Optimization Algorithms' introduces a novel method for automating this discovery process, leading to the creation of a new, efficient algorithm named Lion.



Problem Statement

Discovering efficient optimization algorithms is challenging. Automation can help.

Understanding the Problem

- Deep understanding of machine learning and problem specifics is required.

Designing the Algorithm

- This step involves complex optimization techniques and algorithm design.

Testing and Refining the Algorithm

- The algorithm undergoes extensive testing and refinement, requiring significant time and computational resources.

Challenge - Automation

- The manual and expertise-driven process calls for automation to make algorithm discovery more efficient and accessible.

Objective of the Paper

- The paper presents a method for automated discovery of optimization algorithms using program search.

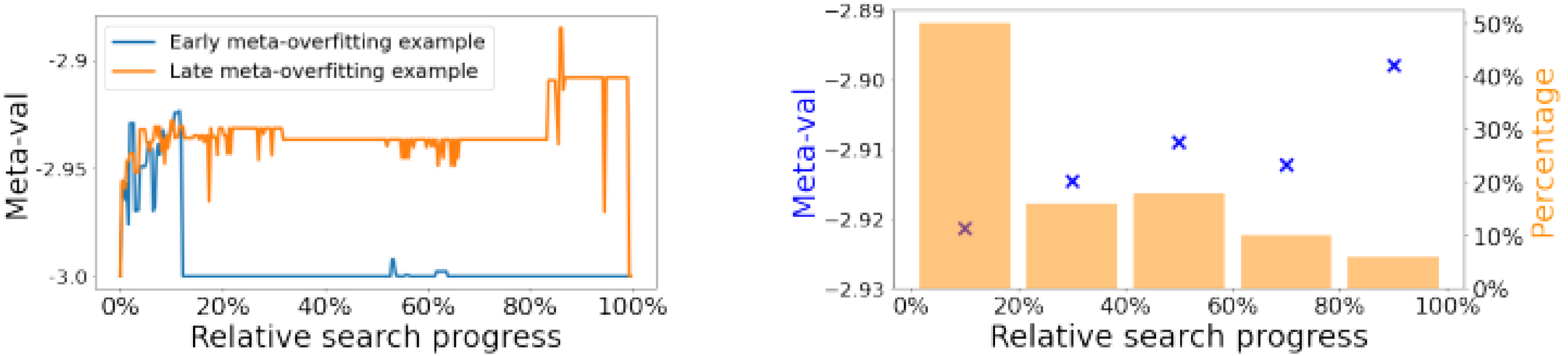
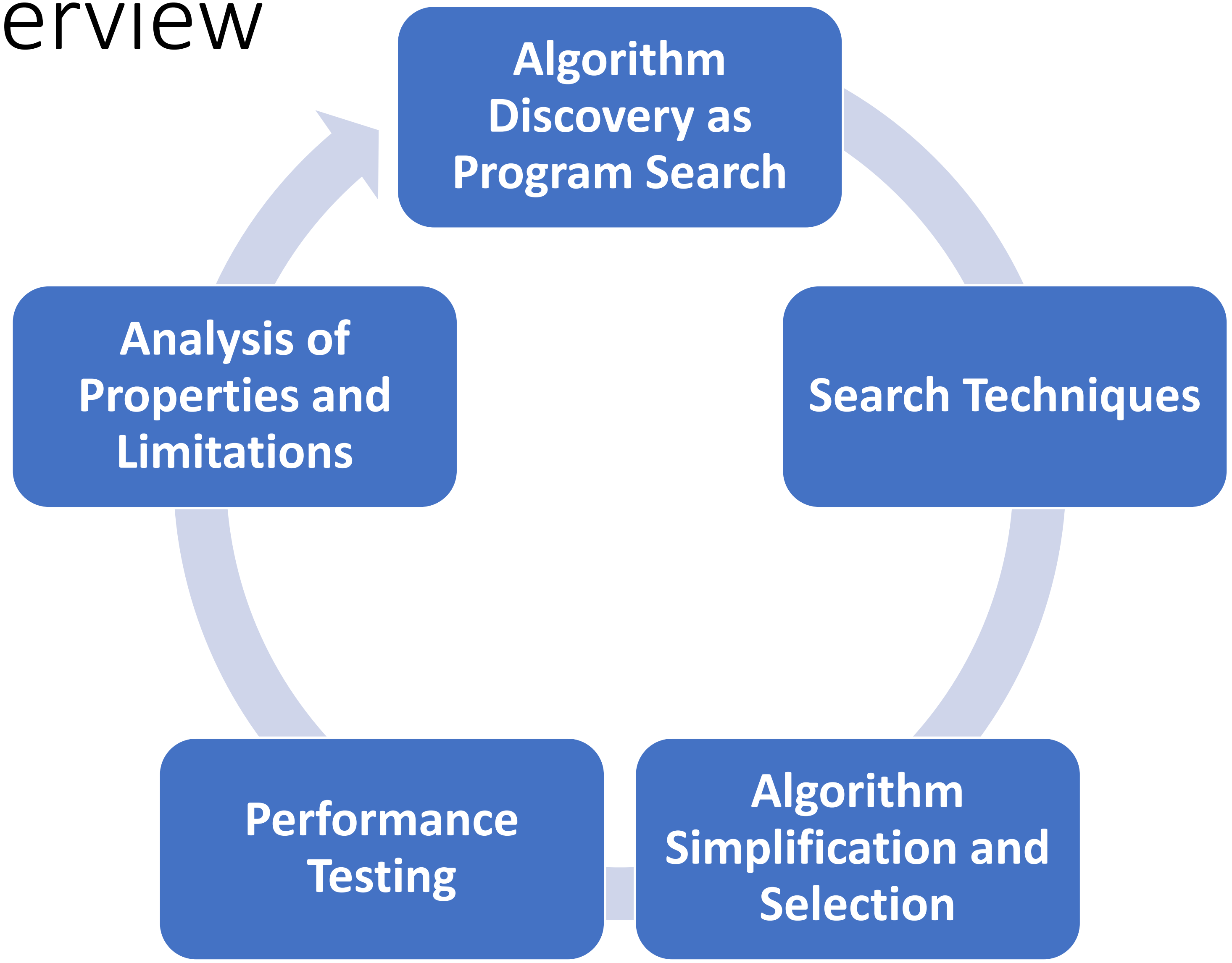


Figure 1 : X. Chen et al., “Symbolic Discovery of Optimization Algorithms,” arXiv:2302.06675v4 [cs.LG], May 2023.

Methodology Overview

- The method involves searching through a space of possible programs to find an efficient optimization algorithm.



Lion - The New Algorithm

- The method led to the discovery of Lion, a simple and efficient optimization algorithm.

Figure 1: **Left**: ImageNet fine-tuning accuracy vs. pre-training cost of ViT models on JFT-300M. **Right**: FID of the diffusion model on 256^2 image generation. We use DDPM for 1K steps w/o guidance to decode image. As a reference, the FID of ADM is 10.94 (Dhariwal and Nichol, 2021).

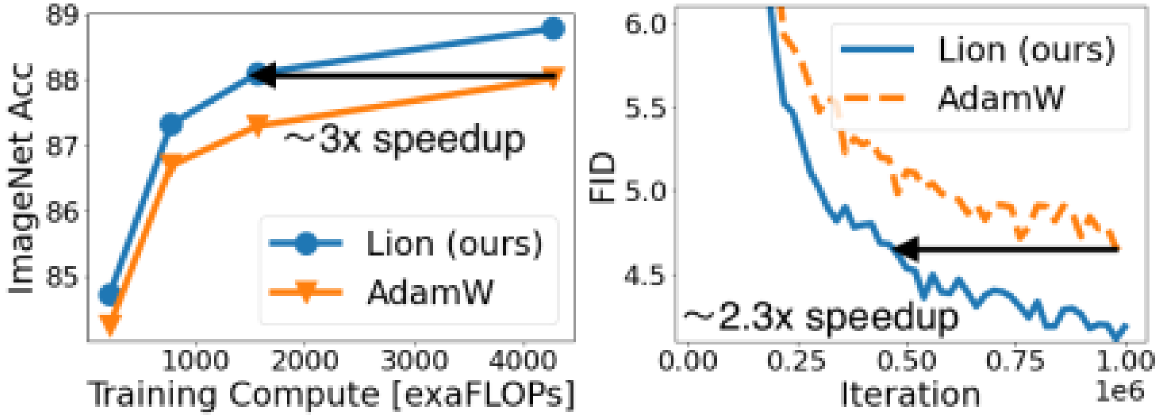
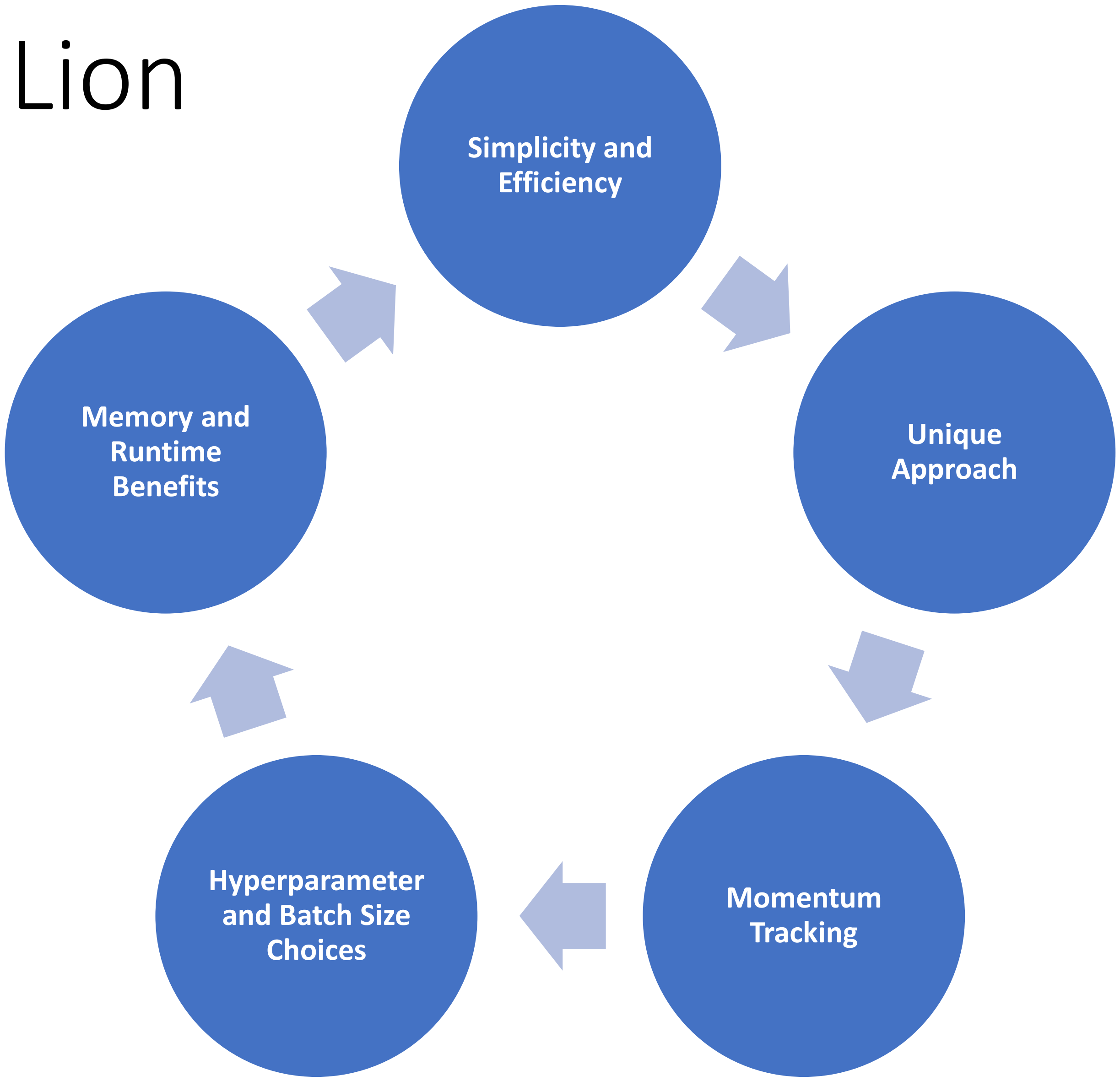


Figure 2 : X. Chen et al., "Symbolic Discovery of Optimization Algorithms," arXiv:2302.06675v4 [cs.LG], May 2023.

Key Features of Lion



Comparison with Adam

Parameter	Lion	Adam
Simplicity and Efficiency	Simpler with fewer hyperparameters. Does not require epsilon and factorization-related hyperparameters.	More complex with more hyperparameters. Requires epsilon and factorization-related hyperparameters.
Learning Rate and Weight Decay	Requires a smaller learning rate and larger weight decay to maintain a similar strength as Adam.	Requires a larger learning rate and smaller weight decay.
Memory and Runtime Benefits	Only saves the momentum, resulting in a smaller memory footprint. Has faster runtime due to its simplicity.	Saves more than just the momentum, resulting in a larger memory footprint. Slower runtime due to complexity.
Performance	More robust to different hyperparameter choices. Performs better on various tasks, especially with large batch sizes.	Less robust to different hyperparameter choices. Performance varies depending on the task and batch size.
Limitations	Performs similarly to Adam on certain tasks. Performance gain decreases when strong augmentations are utilized.	Performs better on certain tasks. Performance gain may vary depending on the task and batch size.

Performance of Lion

- Lion demonstrated outstanding performance across various models and tasks.

Figure 3 : X. Chen et al., “Symbolic Discovery of Optimization Algorithms,” arXiv:2302.06675v4 [cs.LG], May 2023.

Model	#Params	Optimizer	RandAug + Mixup	ImageNet	ReaL	V2
Train from scratch on ImageNet						
ResNet-50	25.56M	SGD	✗	76.22	82.39	63.93
		AdamW		76.34	**82.72**	**64.24**
		Lion		**76.45**	**82.72**	64.02
Mixer-S/16	18.53M	AdamW	✗	69.26	75.71	55.01
		Lion		**69.92**	**76.19**	**55.75**
Mixer-B/16	59.88M	AdamW	✗	68.12	73.92	53.37
		Lion		**70.11**	**76.60**	**55.94**
ViT-S/16	22.05M	AdamW	✗	76.12	81.94	63.09
		Lion		**76.70**	**82.64**	**64.14**
		AdamW	✓	78.89	84.61	66.73
		Lion		**79.46**	**85.25**	**67.68**
ViT-B/16	86.57M	AdamW	✗	75.48	80.64	61.87
		Lion		**77.44**	**82.57**	**64.81**
		AdamW	✓	80.12	85.46	68.14
		Lion		**80.77**	**86.15**	**69.19**
CoAtNet-1	42.23M	AdamW	✓	83.36 (83.3)	-	-
		Lion		**84.07**	-	-
CoAtNet-3	166.97M	AdamW	✓	84.45 (84.5)	-	-
		Lion		**84.87**	-	-
Pre-train on ImageNet-21K then fine-tune on ImageNet						
ViT-B/16_{384}	86.86M	AdamW	✗	84.12 (83.97)	88.61 (88.35)	73.81
		Lion		**84.45**	**88.84**	**74.06**
ViT-L/16_{384}	304.72M	AdamW	✗	85.07 (85.15)	88.78 (88.40)	75.10
		Lion		**85.59**	**89.35**	**75.84**

Limitations of Lion

1. **Performance on Certain Tasks** - Lion's performance gain is minimal on ConvNets and decreases with strong augmentations.
2. **Performance with Small Batch Sizes** - Lion's performance is similar to AdamW for small batch sizes (<64).
3. **Performance on Large Datasets** - Lion's performance is comparable to AdamW on tasks with massive, high-quality datasets.
4. **Limitations of Search** - Lion's search space is biased towards first-order optimization algorithms and lacks functions for advanced second-order algorithms.
5. **Simplistic Program Structure** - Lion's program structure is simplistic and could benefit from incorporating advanced program constructs.

Lion Algorithm: Public Accessibility and Practical Applications

- **Public Availability:** A pytorch implementation of Lion is publicly available (https://github.com/lucidrains/lion-pytorch)
- **Real-World Application:** Lion has been successfully deployed in production systems such as Google's search ads CTR model.
- **Performance in Various Tasks:** Lion has shown outstanding performance across a range of models (Transformer, MLP, ResNet, U-Net, and Hybrid) and tasks (image classification, vision-language contrastive learning, etc.).

Implications of the Research

Advancement of Optimization Algorithms

- Automates discovery process
- Uncovers novel approaches

Improved Training of Deep Neural Networks

- Enhances performance
- Accelerates convergence

Broader Accessibility and Adoption

- Increases accessibility
- Fosters adoption in various domains

Real-World Applications

- Enhances machine learning systems
- Speeds up deployment, reduces costs

Potential for Generalization

- Extends to other domains
- Drives advancements

Future Work

- Further Exploration of Search Space
- Efficiency Improvements
- Application to Other Domains
- Generalization and Transfer Learning

Summary

- Automates the discovery process of optimization algorithms.
- Introduces Lion, a new algorithm for efficient training of deep neural networks.
- Simplifies algorithm discovery, making it more accessible to researchers and practitioners.
- Shows strong performance and potential for generalization.
- Implications include advancements in optimization algorithms, improved deep neural network training, and real-world applications.
- Offers possibilities for broader adoption and enhanced machine learning systems.

References

- X. Chen et al., “Symbolic Discovery of Optimization Algorithms,” arXiv:2302.06675v4 [cs.LG], May 2023.

Q & A