COMPSCI 692S: SYSTEMS FOR ML & ML FOR SYSTEMS

University of Massachusetts Amherst
BE REVOLUTIONARY™

OUR PROGRESS

2015

CNNs surpass human performance on ImageNet classification.

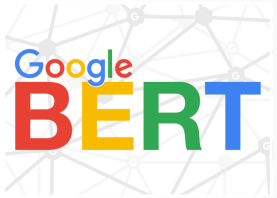
2016

AlphaGo beats a Go World Champion.

2017

AlphaGo Zero masters Go without human knowledge.

2018



BERT is open-sourced.

2019

Microsoft achieves human performance on the GLUE Benchmark.

2020

?

OUR PROGRESS

2015

CNNs surpass human performance on ImageNet classification.

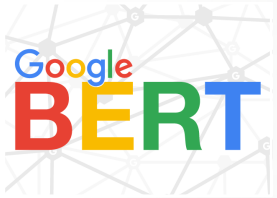
2016

AlphaGo beats a Go World Champion.

2017

AlphaGo Zero masters Go without human knowledge.

2018



BERT is open-sourced.

2019

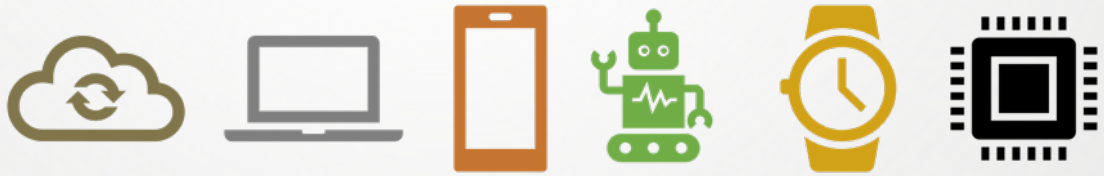
Microsoft achieves human performance on the GLUE Benchmark.

2020

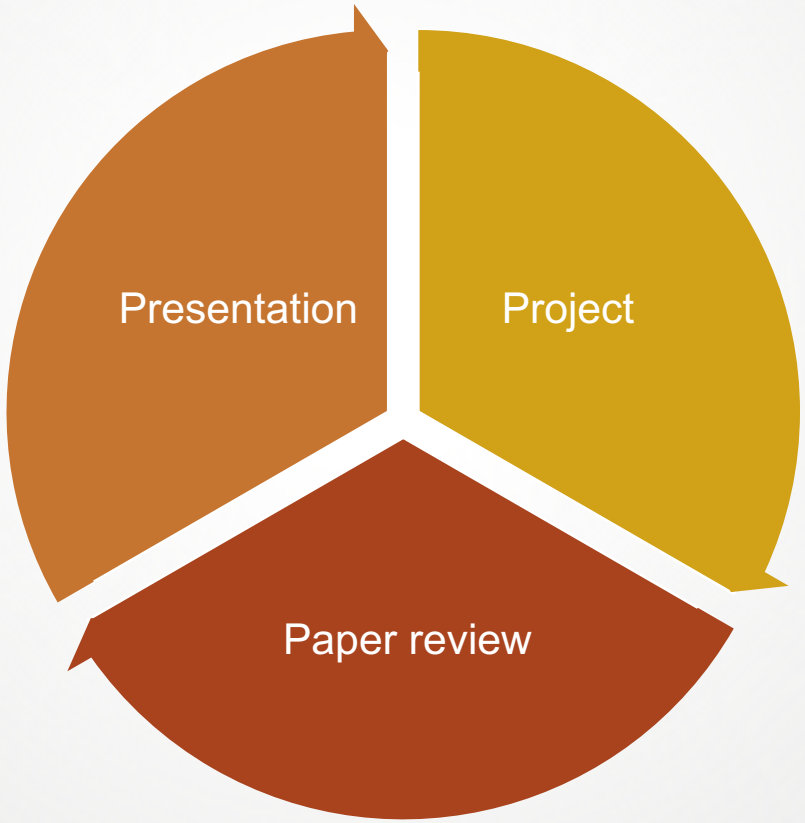
ABOUT THIS SEMINAR

ML Applications  ...

Adopt ML into real-world applications → design and implement ML systems.

Various Platforms  ...

COURSE STRUCTURE



COURSE STRUCTURE

- **Classes (online via zoom, Wed. 11:15 AM – 1:15 PM):**
  - Zoom link: https://umass-amherst.zoom.us/j/92411414412?pwd=OGxnM0trLzR5V2ZQMlJlbzJhbVVVdz09
  - 1 presentation per class by a presentation group
  - Each presentation group has **2-3 students**
  - Each presentation group covers **an area and 3 papers in the area**.
  - Each student in **at least one** presentation group.
- **Paper review before each class:**
  - Everyone reads the 3 papers and submits paper reviews using review forms
  - Deadline: Tuesday, 11:59 PM, before the papers are presented in class.
- **A project If you are in the 3 credit section**
  - A project group: 1-3 students
  - More work will be expected from groups with more people

PAPER PRESENTATION

- **During presentation**
  - Give context, present the problem (not solutions!): 15 mins
  - Present papers: 40 mins
  - Discussion: comparison, strengths, weaknesses, insights: 10 mins
- **After presentation Q&A: 10 mins**
- **Everyone joins at least 1 presentation group**
  - Spreadsheet: https://docs.google.com/spreadsheets/d/1tCGYAVOFP7pyR0tbwaq5k_QUbF_QZtlSQXnqgFC_-hM/edit?usp=sharing
- **I will announce papers to read.**
  - If your presentation group wish to change topics or papers to read, please let me know at least 1 week before.

Paper-Review-09-09

Topic: Efficient Training II

* Required

Student ID *

Your answer

First Name *

Your answer

Last Name *

Your answer

Email *

Your answer

Paper ID *

Paper #1

Summary *

Your answer

Strengths (3 or more) *

Your answer

Weaknesses (3 or more) *

Your answer

Lessons learned (3 or more) *

Your answer

Submit

THINK THESE QUESTIONS WHEN REVIEWING

- Is the paper well-motivated? What problem does it address, and is it an important problem?
- Does the paper significantly advance the state of the art or break new ground?
- What are the paper's key insights?
- What are the paper's key scientific and technical contributions?
- Does the paper credibly support its claimed contributions?
- What did you learn from the paper?
- Does the paper clearly establish its context with respect to prior work? Does it discuss prior work accurately and completely? Are comparisons with previous work clear and explicit?
- Does the paper describe something that has actually been implemented? If so, has it been evaluated properly? Is it publicly available so that these results can be verified?
- What impact is this paper likely to have (on theory & practice)?

Source: https://emeryblogger.com/2018/03/22/reviewing-guidelines-for-program-committee-members/

PROJECTS

- **We will have project weekly meeting:**
  - Please vote using this doodle pool: https://doodle.com/poll/ukr6ffbzawxia42s
  - **DDL: Today, 8/26/2020**
- **Tentative Project Timeline:**
  - Identify a Sys4ML or ML4sys problem: either your proposal or my choice (Sept. 20)
  - Evaluate the existing solutions and propose improvements (Nov. 17)
  - Final presentation: poster or slides (Nov. 18)
  - Final report and code (Dec. 4)
    - Report in MLSys'21 format: https://mlsys.org/Conferences/2021/CallForPapers

LOGISTICS

- **During a class, put video on if you can**
  - We allow people to interrupt and ask questions at **anytime**!
  - Please mute yourself when not talking to avoid unintended interruptions.
- **We may use other time and zoom link for invited speakers!**
  - Wednesday, 9/23, 11:15 AM – 1:15 PM: Zhihao Jia
    - Talk title: Automated Discovery of Machine Learning Optimizations
  - Friday, 11/13, 12:00 PM – 1:00 PM: Tianqi Chen
    - Talk title: TVM: An automated deep learning compiler

LOGISTICS CONT.

- **All communications are through Piazza**
  - Ask questions, answer others' questions, find teammates
  - Signup: piazza.com/umass/fall2020/compsci692s01 (access code: mlsys2020fall)
  - Note: please use complete sentences. Avoid all caps.

BE REVOLUTIONARY™

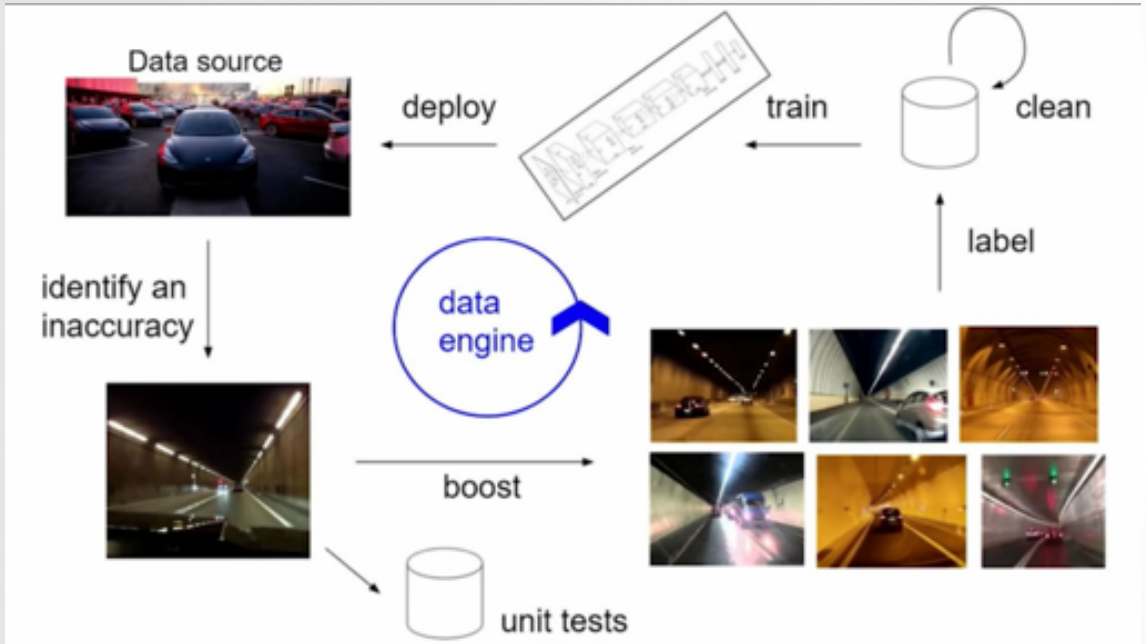
SYSTEMS FOR ML

University of Massachusetts Amherst

MACHINE LEARNING IS POPULAR BECAUSE…

However,
designing and implementing the systems that support ML models in real-world deployments remains a significant obstacle.

EXAMPLE: SELF-DRIVING CARS



Questions:

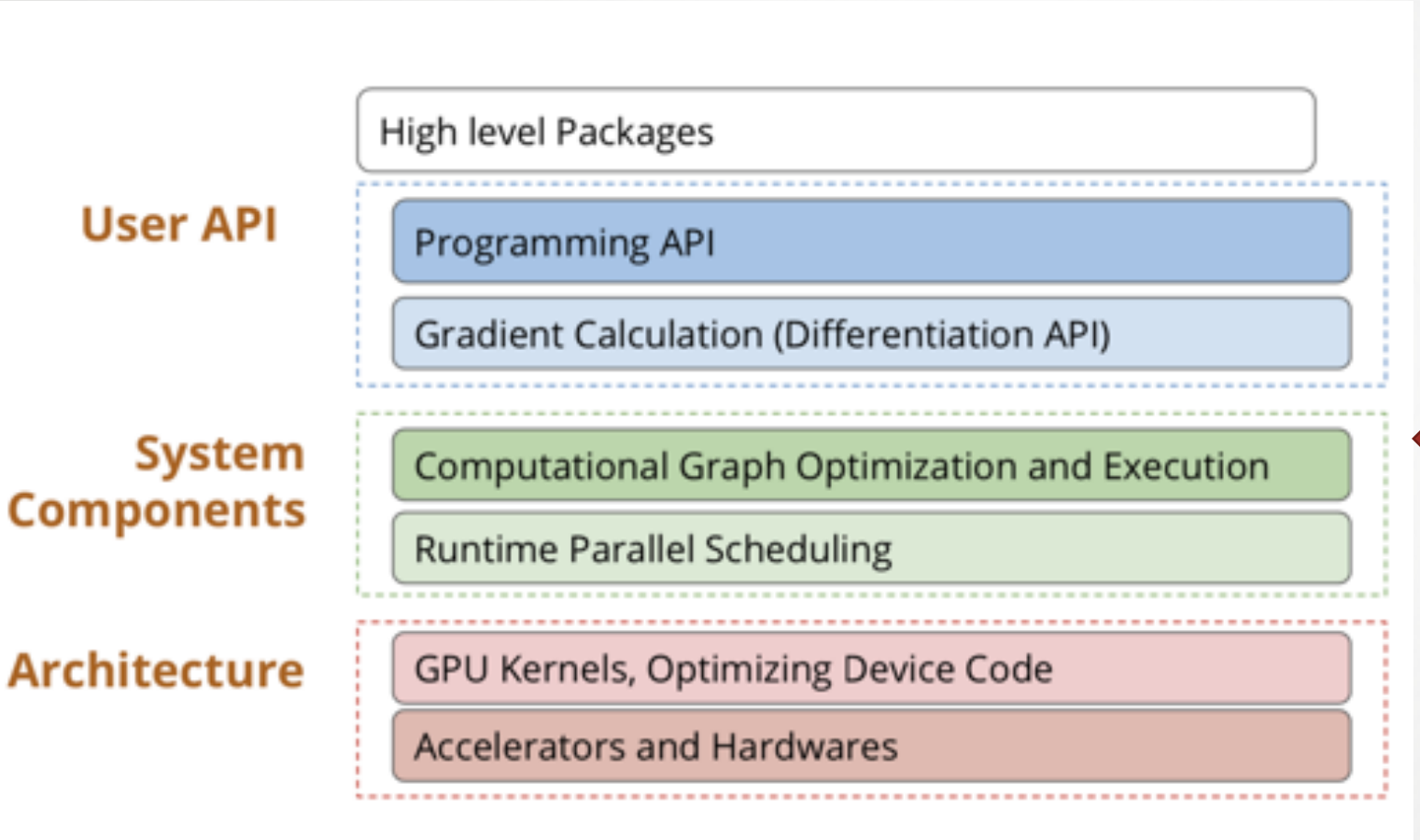
- **Data labeling is expensive.**
- **Multiple models v.s. 1 model for multiple tasks: road sign, traffic light, object detection, etc.**
- **Need 99.99% accuracy, not 97%.**
- **How to meet the latency requirement?**
- **…**

This image is cropped from: https://slideslive.com/38917690/multitask-learning-in-the-wilderness

MAIN QUESTIONS

- **How should software systems be designed to support the ML lifecycle?**
  - Enable users to quickly “program” the modern ML stack
  - Enable developers to define and measure ML models, architectures, and systems
  - Support efficient development, monitoring, debugging, etc. of ML applications
- **How should hardware systems be designed for machine learning?**
  - Develop specialized hardware for training and deploying ML models
  - Discover new tradeoffs wrt. accuracy, performance, robustness etc.
  - Design distributed systems to support ML training and serving
- **How should ML systems be designed to satisfy metrics beyond predictive accuracy?**
  - Support resource-constrained devices (e.g., power, latency, memory limit)
  - Support privacy and security guarantees
  - Increase the accessibility of ML to a broader range of users

TYPICAL DEEP LEARNING SYSTEM STACK



Require innovations in:

- Algorithms
- Programming Language
- Software Engineering
- Compilers
- Architectures

Image from: http://dlsys.cs.washington.edu/pdf/lecture3.pdf

EXAMPLE: WOOTZ

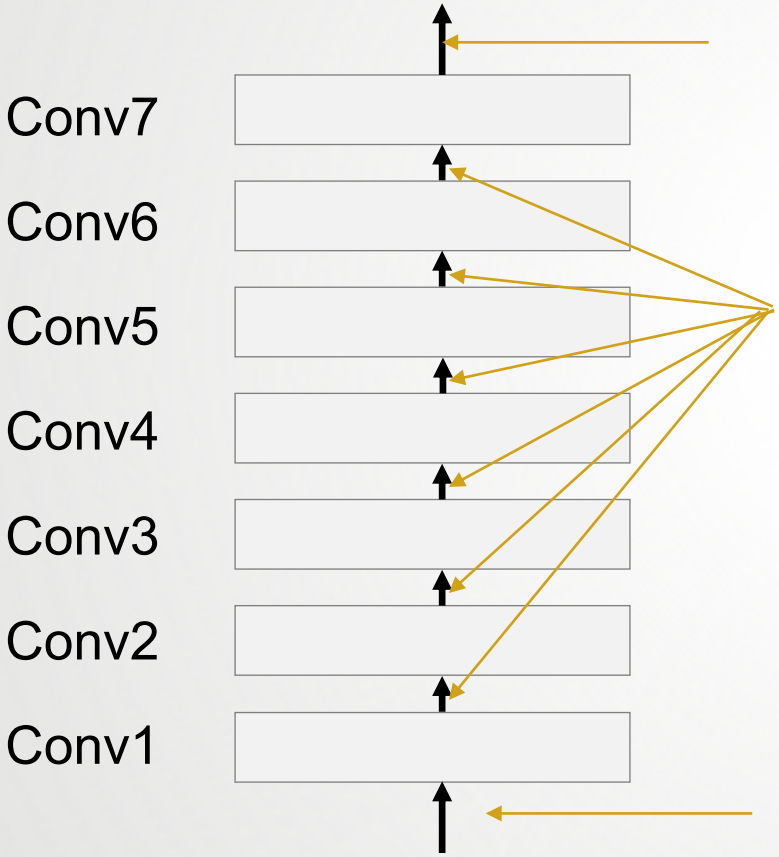
Wootz: A Compiler-Based Framework for Fast CNN Pruning Via Composability

- Composability-based CNN Training to accelerate model design
- Compiler support to ease implementation

Require innovations in:

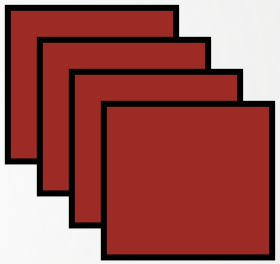
- **Algorithms**
- Programming Language
- Software Engineering
- **Compilers**
- Architectures

CONVOLUTIONAL NEURAL NETWORK

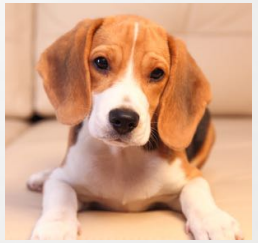


Output: Dog or Cat?

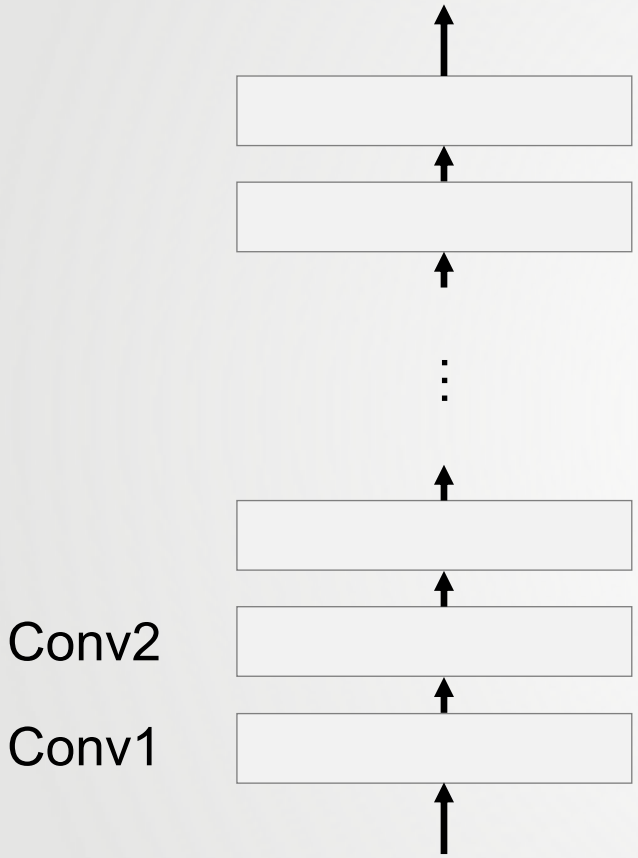
Intermediate Outputs (Activation Maps):



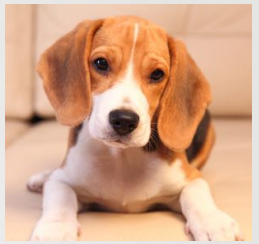
Input:



A CNN Model

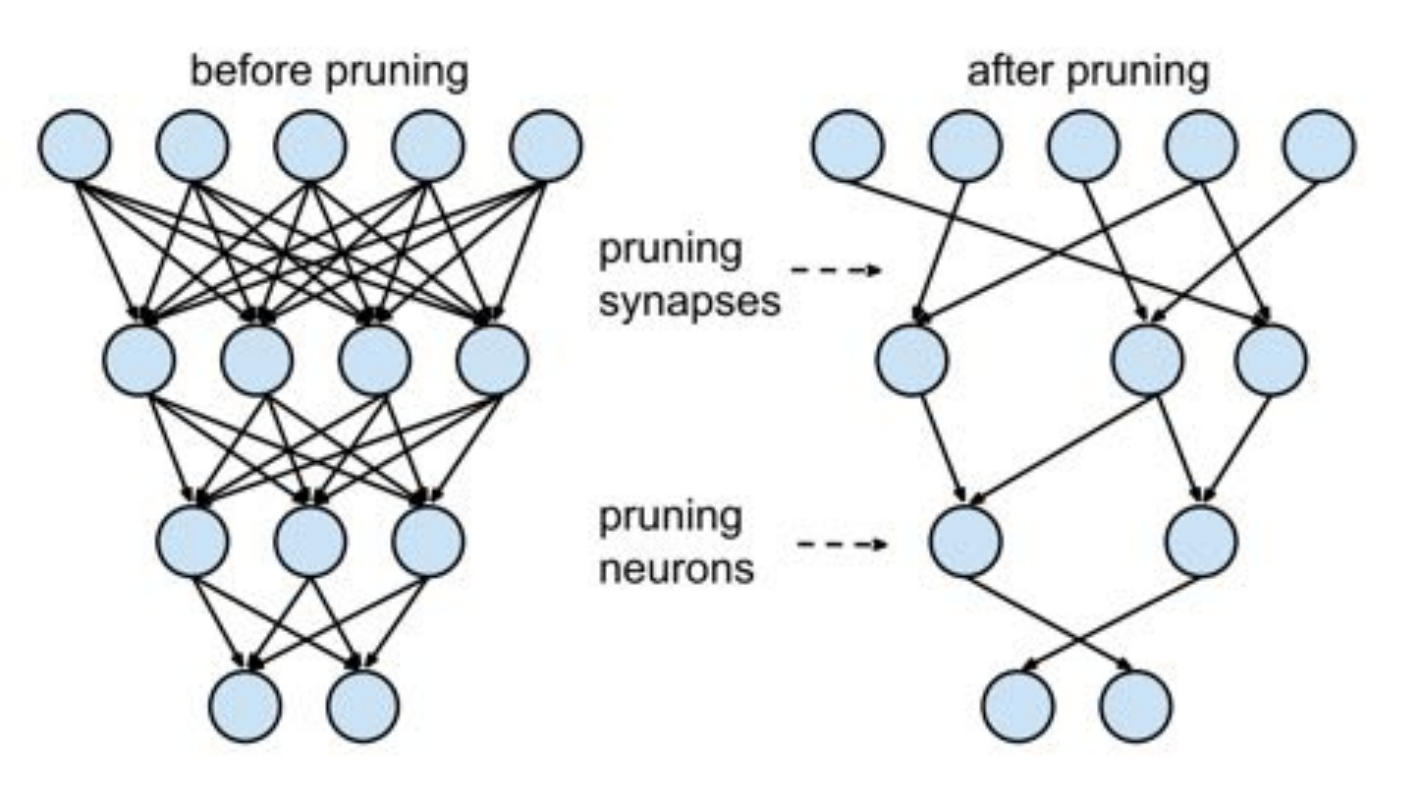


Too large to fit into your phone?



CNN Pruning

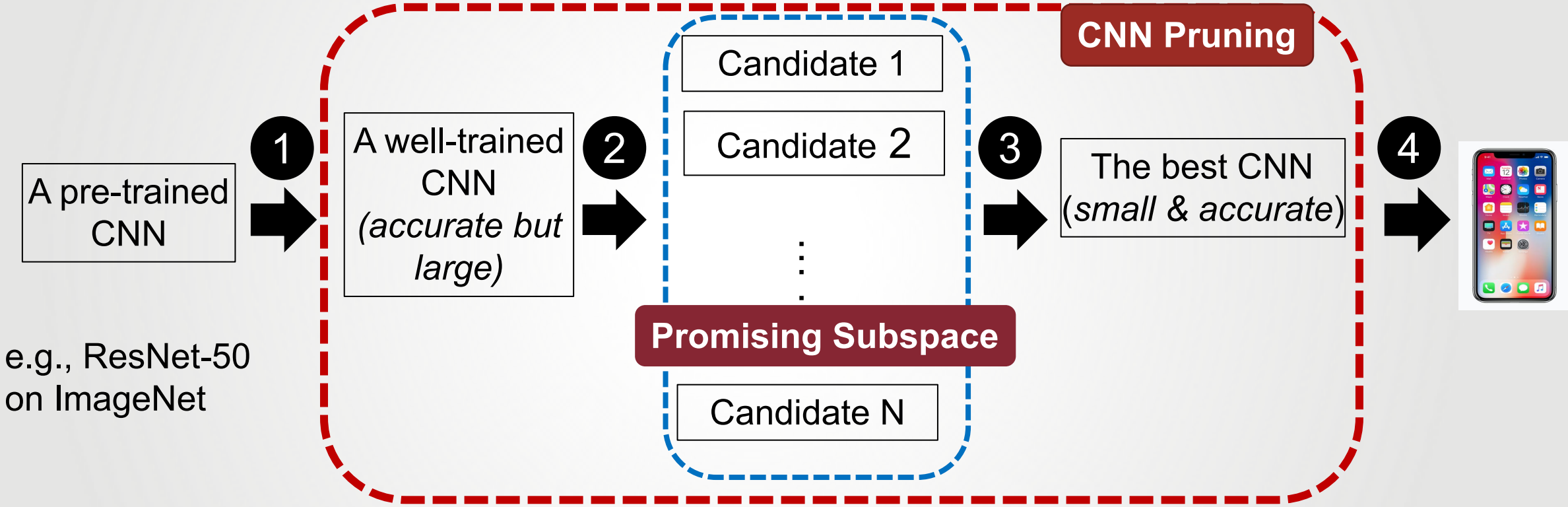
PRUNING



Benefits of CNN pruning:

- Reduce model size
- Reduce inference latency
- Improve generalization ability

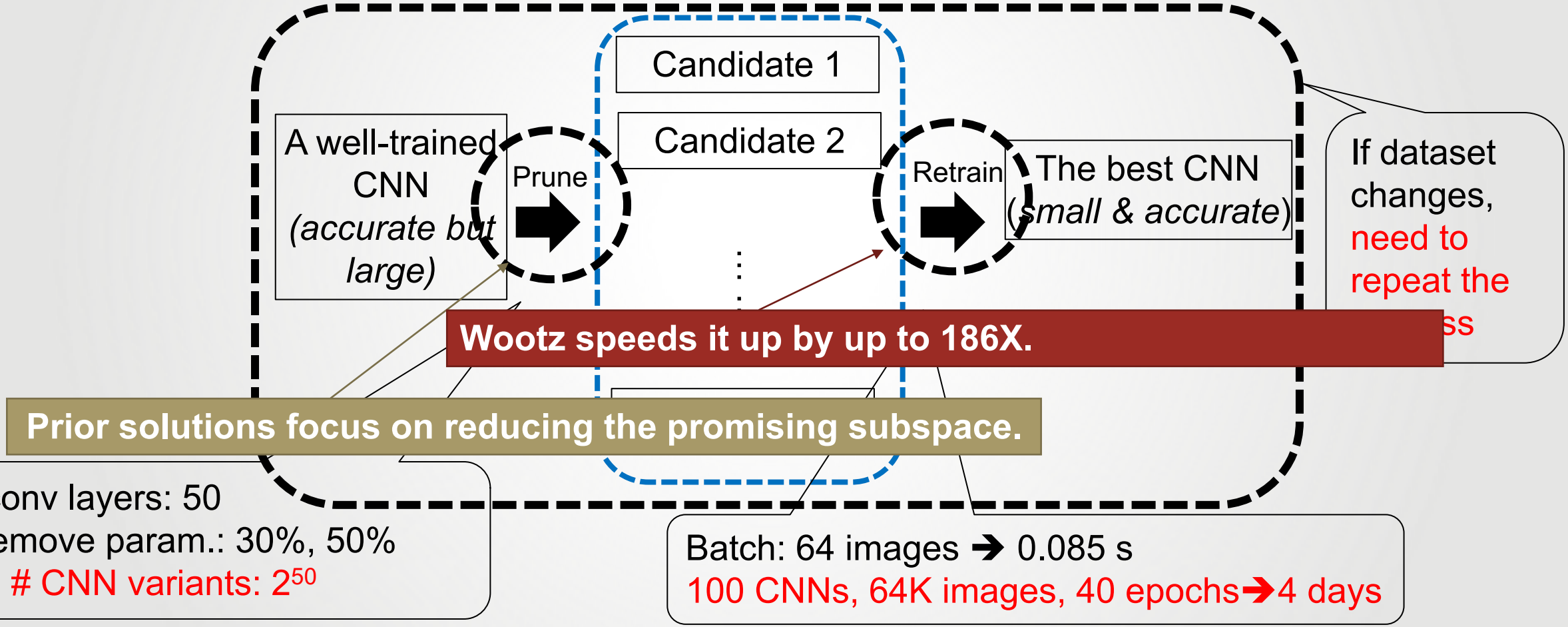
PRUNING PROCESS



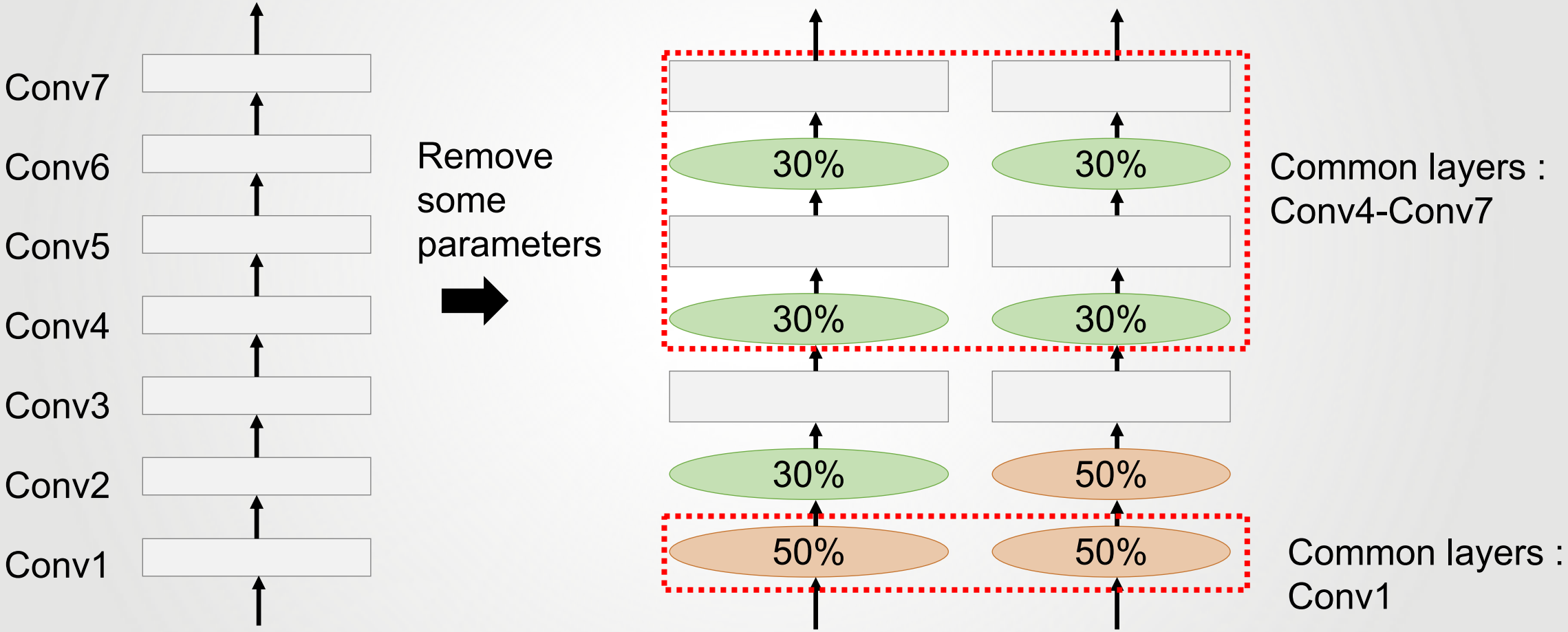
e.g., ResNet-50 on ImageNet

1. Finetune on the dog dataset
2. Prune by removing some parameters
3. Retrain and choose the best candidate
4. Deploy

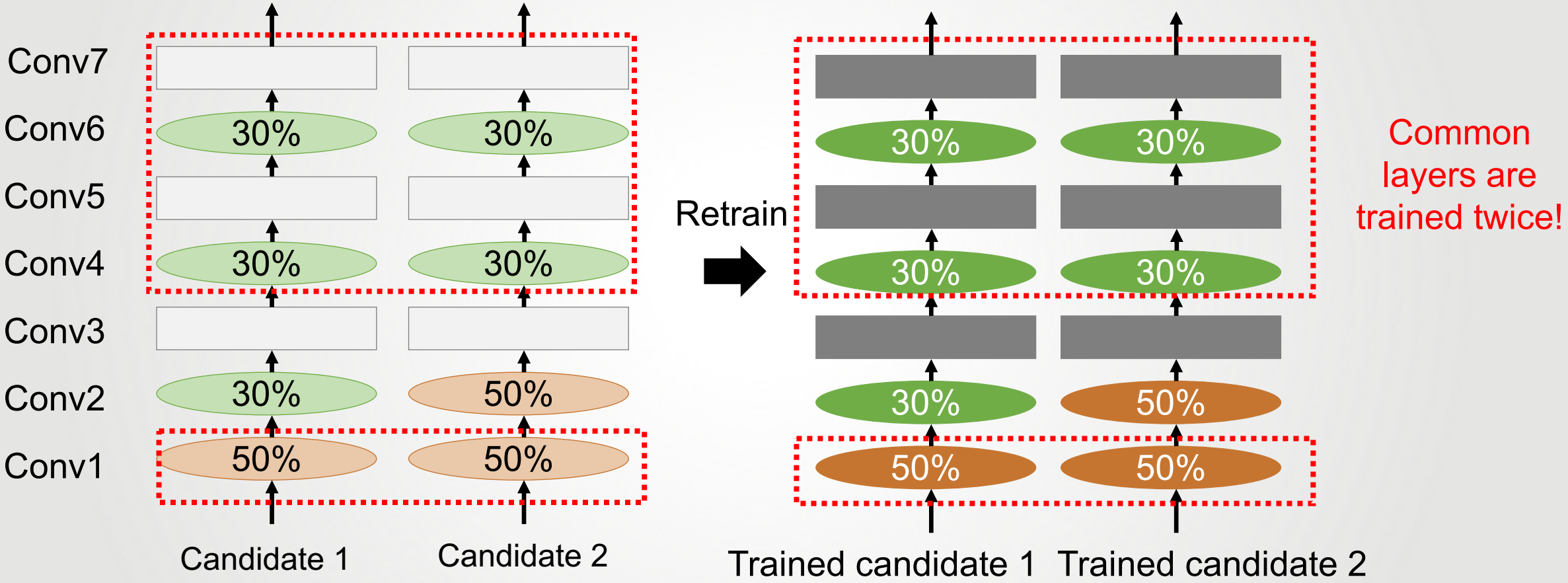
PRUNING IS TIME-CONSUMING



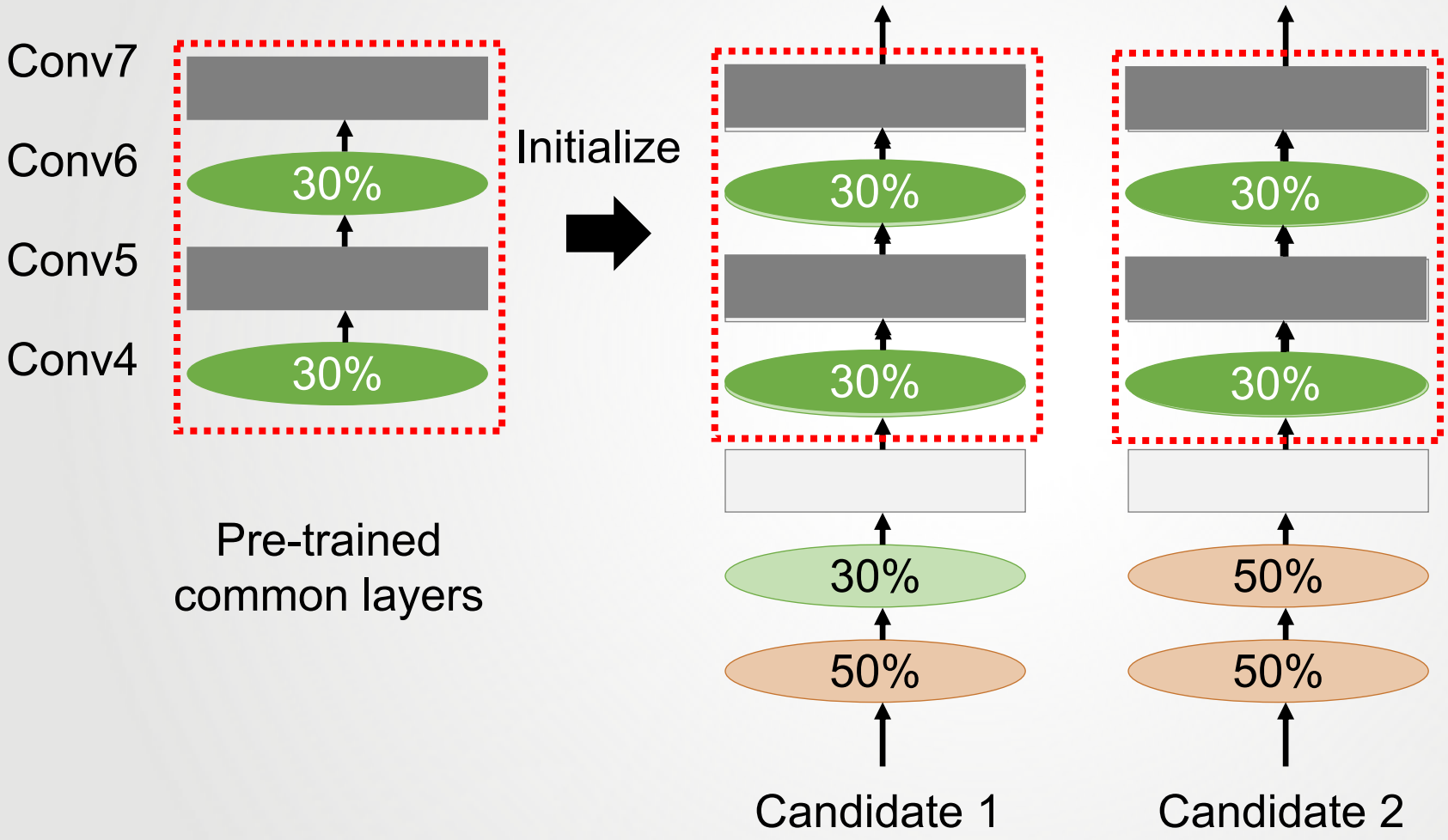
Pruned CNN variants often have several layers in common (common: same amount of parameters).



Prior methods: train from scratch and test for accuracy.

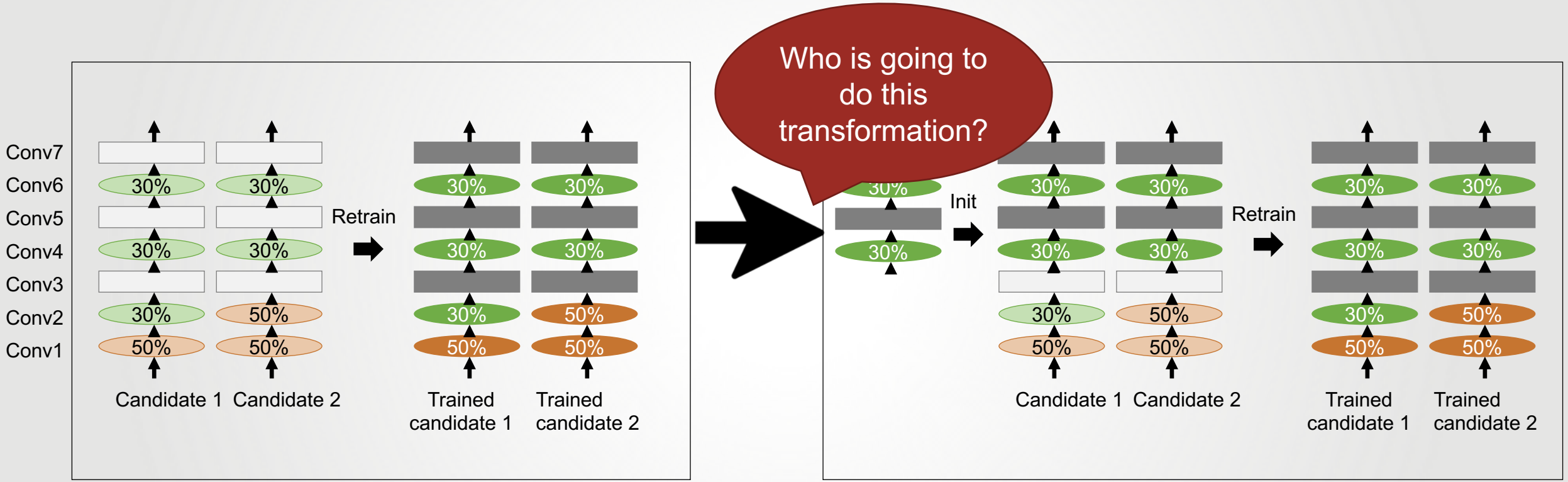


Basic idea: composability? Reuse?

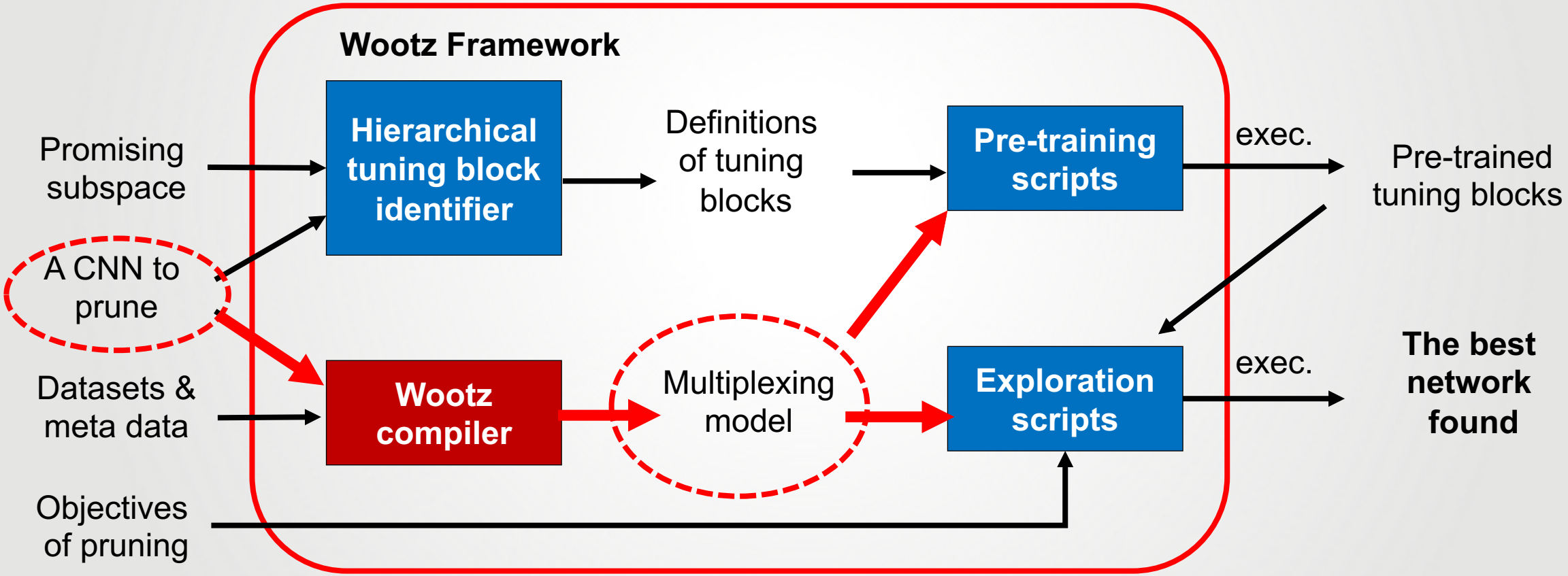


Save training time of pruned candidates 1&2?

Algorithm Innovation: Composability-based CNN Training



Compiler Support: Wootz compiler and scripts automatically materialize the composability-based CNN pruning for an arbitrary CNN.



KEY TAKEAWAYS FROM WOOTZ

ML Algorithms

Support
(e.g., ease of programming)

Inspire
(e.g., reuse principle)

Apply
(e.g., compiler optimization)

Programming Systems (compilers, runtime)

Multi-task learning
(e.g., model co-run, model packing)

More opportunities:

- Algorithms
- Programming language
- Software Engineering
- Compilers
- Architectures
- …

RESEARCH TOPICS FROM MLSYS'21

- Topics of interest include, but are not limited to:
  - **Efficient model training, inference, and serving**
  - **Distributed and parallel learning algorithms**
  - **Privacy and security for ML applications**

  } This seminar

  - Testing, debugging, and monitoring of ML applications
  - Fairness, interpretability and explainability for ML applications
  - Data preparation, feature selection, and feature extraction
  - ML programming models and abstractions
  - Programming languages for machine learning
  - Visualization of data, models, and predictions
  - Specialized hardware for machine learning
  - Hardware-efficient ML methods

Source: https://mlsys.org/Conferences/2021/CallForPapers

BE REVOLUTIONARY™

ML FOR SYSTEMS

University of Massachusetts Amherst

WHY ML FOR SYSTEMS

- **Systems use heuristics to decide**
  - Data structures (e.g., B-tree)
  - Data representations (e.g., row v.s. column stores)
  - Memory allocation (e.g., which part of memory to allocate data)
  - System configurations
  - …
- **Heuristics can fall short**
  - Based on assumptions about the workload, which might be wrong
  - Cannot consider complex interactions

ML FOR SYSTEMS

- **Replace these heuristics with learned components**
- **Becoming popular in**
  - data management: learn to index
  - memory management: learn to prefetch data from memory to cache
  - programming language: verification
  - compiler: learn to determine transformation orders
  - …

RESEARCH TOPICS FROM ML FOR SYSTEMS

- Supervised, unsupervised, and reinforcement learning research with applications to:
  - Systems Software
  - Runtime Systems
  - Distributed Systems
  - Security
  - Compilers, data structures, and code optimization
  - Computer architecture, microarchitecture, and accelerators
  - Circuit design and layout
  - Interconnects and Networking
  - Storage
  - Datacenters
- Representation learning for hardware and software
- Optimization of computer systems and software
- Systems modeling and simulation
- Implementations of ML for Systems and challenges
- High quality datasets for ML for Systems problems

This seminar?

Source: http://mlforsystems.org/call_for_papers.html

REFERENCE

- Ratner, Alexander, et al. "MLSys: The New Frontier of Machine Learning Systems." arXiv preprint arXiv:1904.03257 (2019).
- Guan, Hui, Xipeng Shen, and Seung-Hwan Lim. "Wootz: a compiler-based framework for fast CNN pruning via composability." *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*. 2019.

GRADING

1-Credit Section

- **50% Participation**
  - Presentation
  - Discussion
- **50% Reviews**

3-Credit Section

- **20% Participation**
- **20% Reviews**
- **60% Course Project**
  - Final presentation: Slides or Posters (20%)
  - Report in MLSys'21 format (20%)
  - Code (20%)