上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Week 7 Extension: Structured Pruning

Tao Huang

John Hopcroft Center, School of Computer Science, Shanghai Jiao Tong University

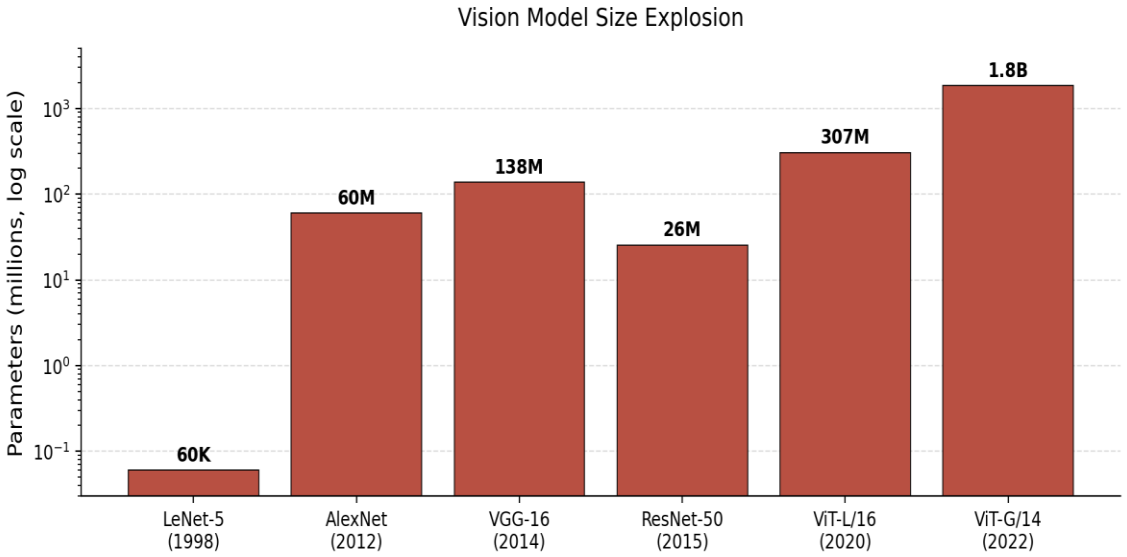
https://taohuang.info/cs3317

https://oc.sjtu.edu.cn/courses/89538

AI tools assisted in generating some figures in these slides. All such content has been reviewed, and the instructor is responsible for its accuracy.

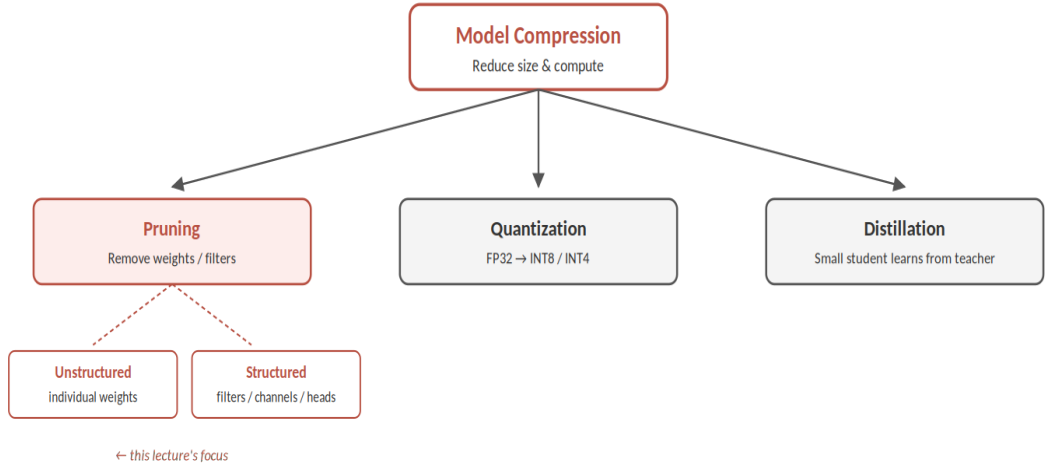
Why Make Models Smaller?

- **Vision models have grown 30000x in 25 years**
  - From 60K parameters (LeNet) to 1.8B (ViT-G)
- **But deployment targets are often constrained:**
  - Mobile phones, edge devices, cars, drones
  - Memory, energy, latency all limited
- **Goal: shrink model without losing accuracy**
  - Faster inference, less memory, lower energy



Three Axes of Model Compression

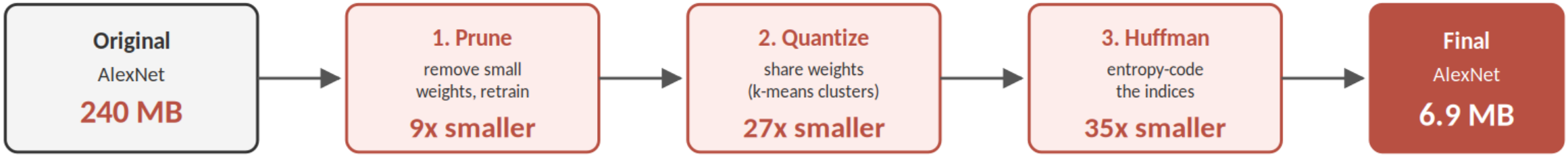
- **Pruning: remove weights or entire filters**
  - Exploit redundancy in learned networks
  - Today's focus
- **Quantization: reduce numerical precision**
  - FP32 to INT8, INT4, or even binary
- **Distillation: train a small student model**
  - Student learns from a large teacher
- **These techniques compose: prune + quantize + distill**



Deep Compression

Han et al. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. ICLR 2016

- **Stage 1 - Pruning**
  - Remove weights with |w| < threshold, then fine-tune
- **Stage 2 - Weight sharing via k-means**
  - Cluster weights, store only cluster index + codebook
- **Stage 3 - Huffman coding**
  - Entropy-code the indices using their distribution
- **Result: 35x smaller AlexNet, no accuracy loss**
  - Showed pruning is a first-class compression tool



Unstructured vs Structured Pruning

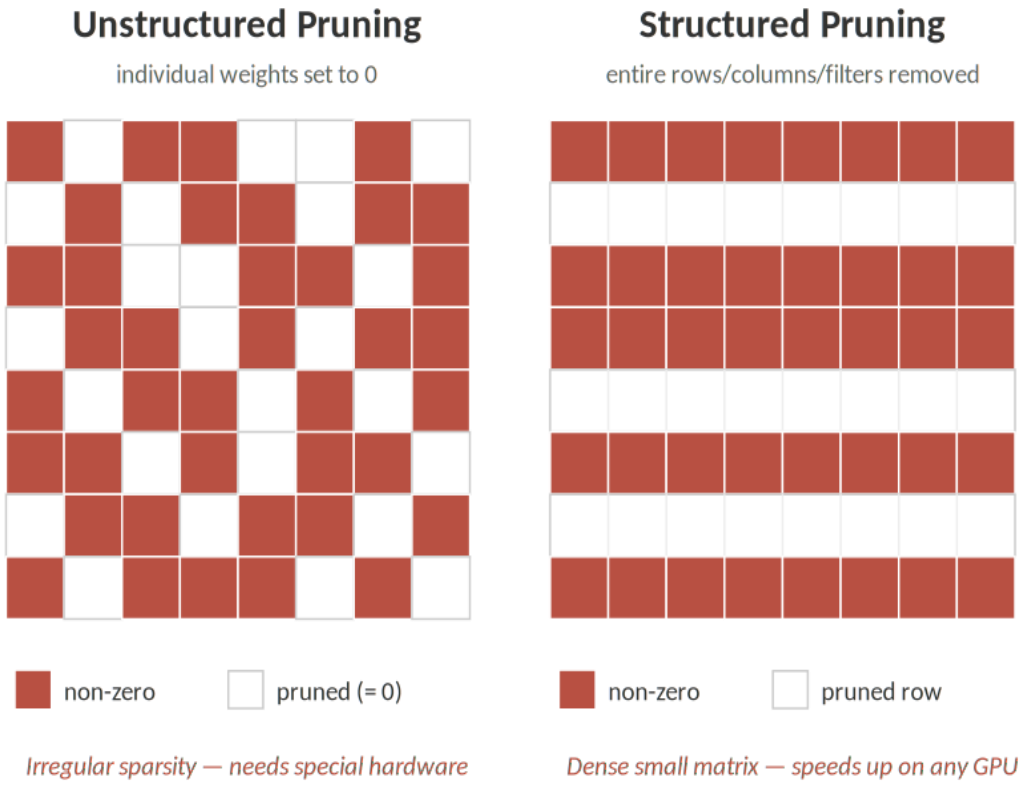
Unstructured: individual weights zeroed

- Highest theoretical compression ratio
- But sparse matmul is slow on GPUs
- Needs special kernels or hardware

Structured: entire rows, filters, or heads removed

- Produces a smaller DENSE network
- Direct speedup on any hardware
- Usually the deployment choice

Key insight: hardware drives the granularity



Pruning Filters for Efficient ConvNets

Li et al. Pruning Filters for Efficient ConvNets. ICLR 2017

Idea: rank filters by a simple importance score

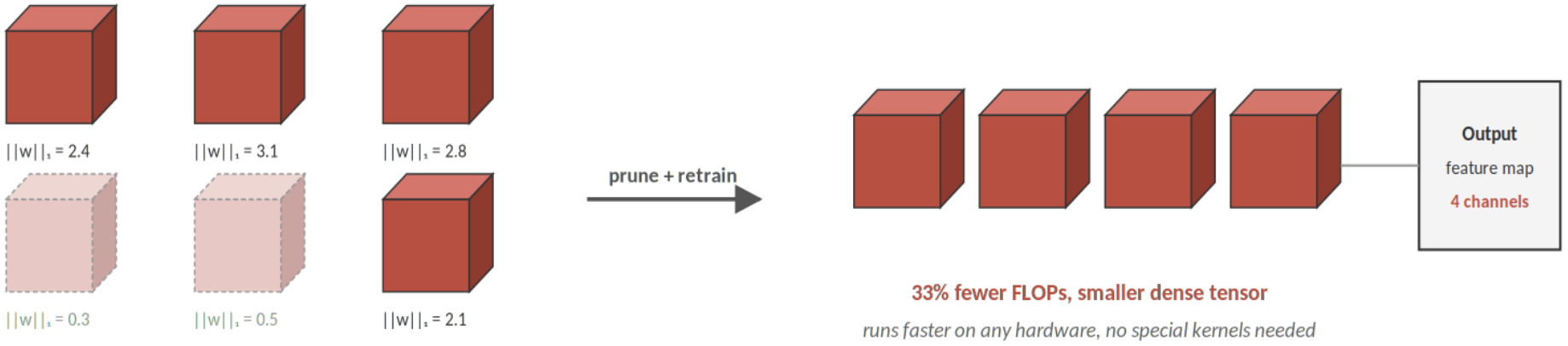
- L1 norm of filter weights: sum of |w|
- Low-norm filters produce weak activations

Algorithm: prune bottom k%, then fine-tune

- One-shot or iterative

Why it works in practice

- Output is still a dense conv layer
- Standard kernels, no special hardware
- Directly reduces FLOPs and parameters



Rethinking the Value of Network Pruning

Liu et al. Rethinking the Value of Network Pruning. ICLR 2019

Surprising finding: the weights don't matter

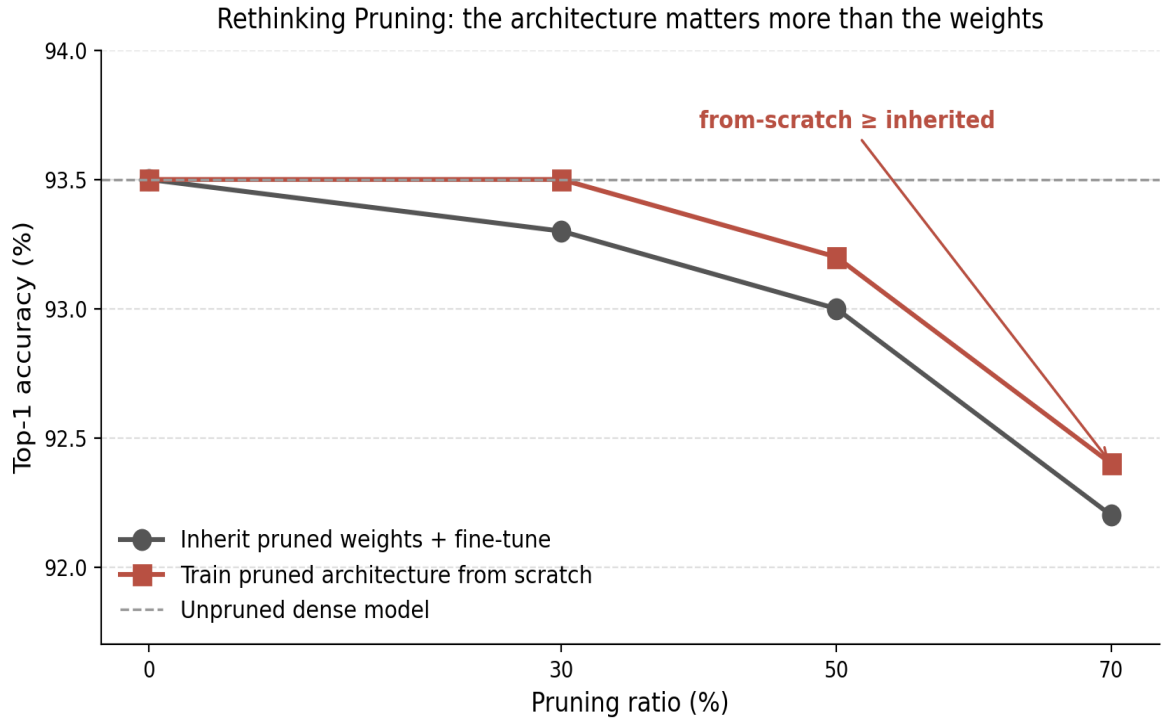
- Train pruned architecture FROM SCRATCH with random init
- Matches, sometimes beats, inheriting pruned weights

Pruning is implicit architecture search

- The VALUE of pruning is finding a good small architecture
- The learned weights are a side effect

Implication

- Pruning bridges to NAS (Week 5 extension)
- Channel counts matter more than weight values



The Lottery Ticket Hypothesis

Frankle & Carbin. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. ICLR 2019

Counterpoint to Liu et al.

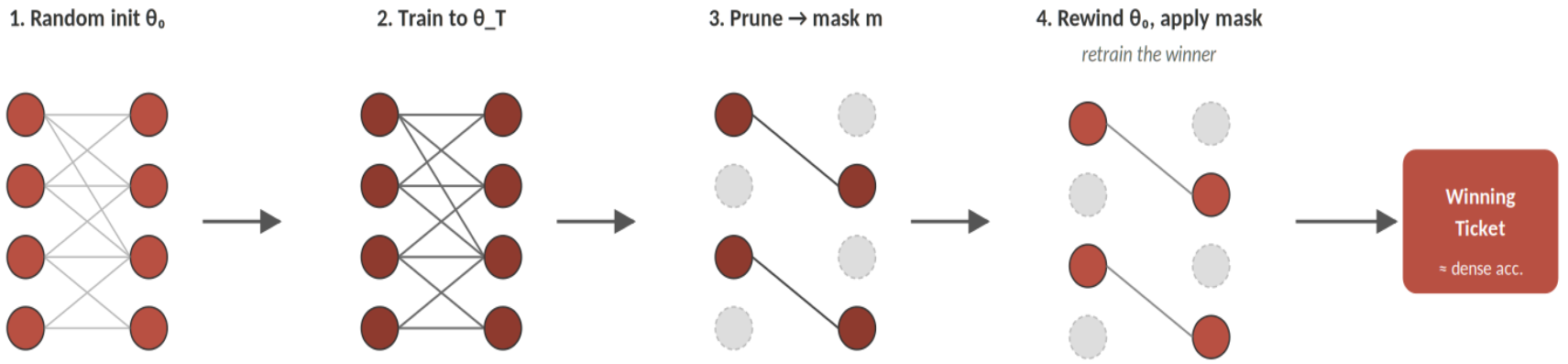
- The winning subnetwork needs ITS OWN original init
- Retraining with a new random init fails

Procedure

- Train dense, prune, rewind surviving weights to init
- Retrain sparse subnet, reach dense accuracy

Why it matters

- Training dynamics depend on initialization, not just architecture
- Opens questions on when and why SGD works



Pruning Transformers and LLMs

Frantar et al. SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot. ICML 2023

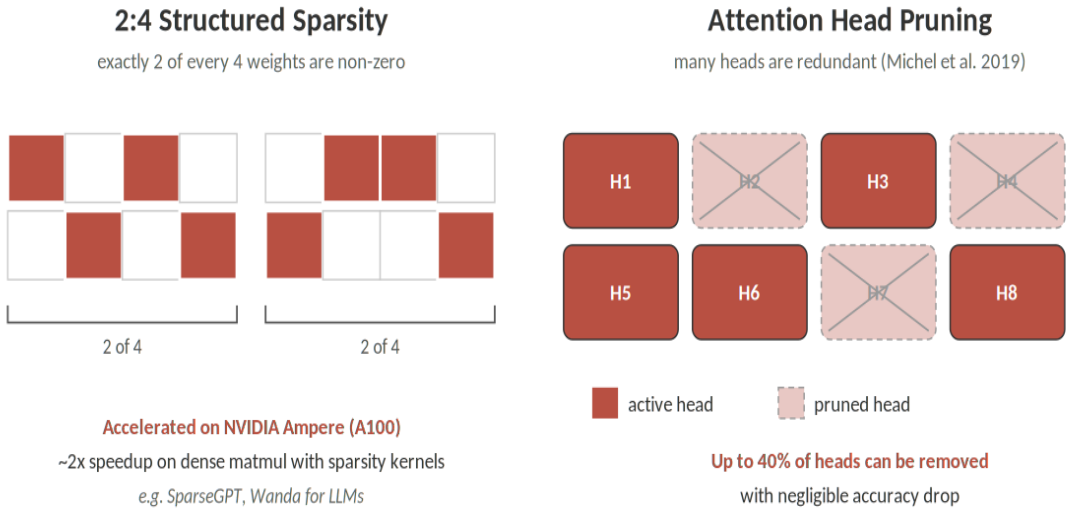
Vision pruning ideas now drive LLM compression

2:4 Structured Sparsity

- Exactly 2 of every 4 weights are non-zero
- Hardware-accelerated on NVIDIA Ampere / Hopper
- SparseGPT, Wanda: one-shot LLM pruning

Attention head pruning

- Many heads are redundant (Michel et al.)
- Up to 40% of heads removable at test time



Takeaway: structured sparsity dominates large-model era

Practical Guide: Which Pruning Method?

Choose based on deployment hardware

Commodity GPU / CPU

- Structured filter or channel pruning (Li et al.)

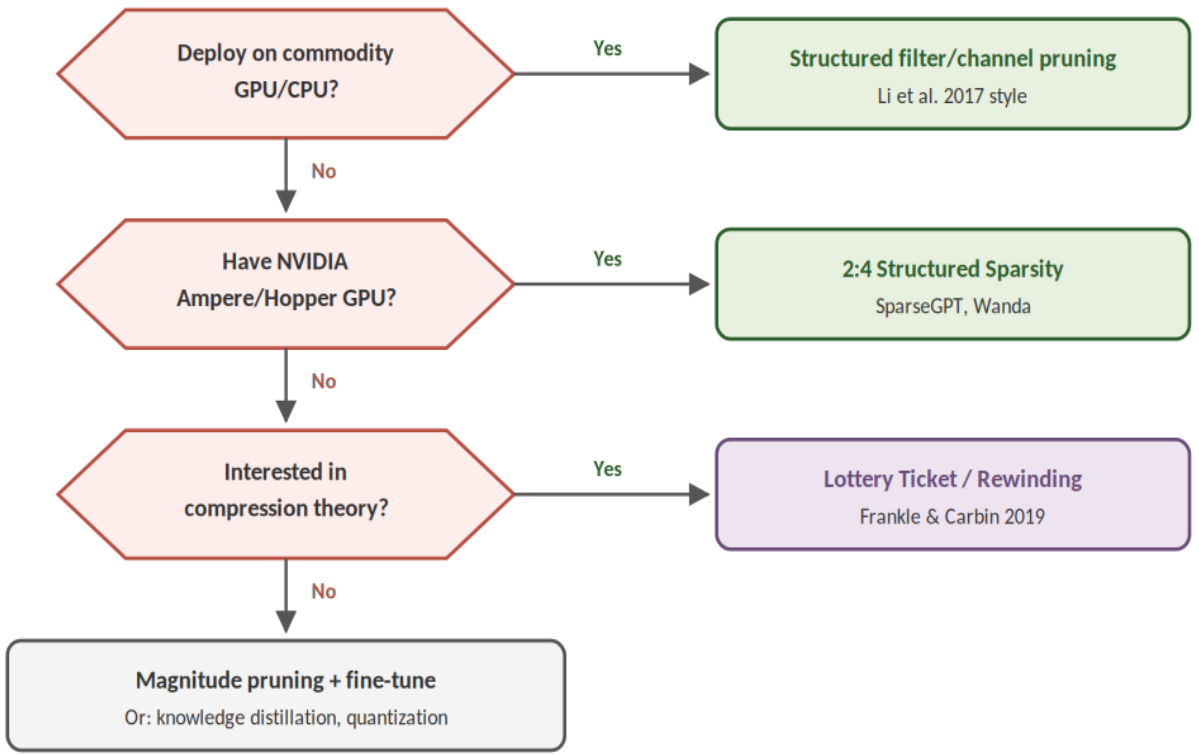
NVIDIA Ampere / Hopper

- 2:4 structured sparsity with hardware kernels

Research / theory

- Lottery tickets, weight rewinding

Libraries: torch.nn.utils.prune, DeepSparse, TensorRT



Summary: The Pruning Landscape

Two axes of progress

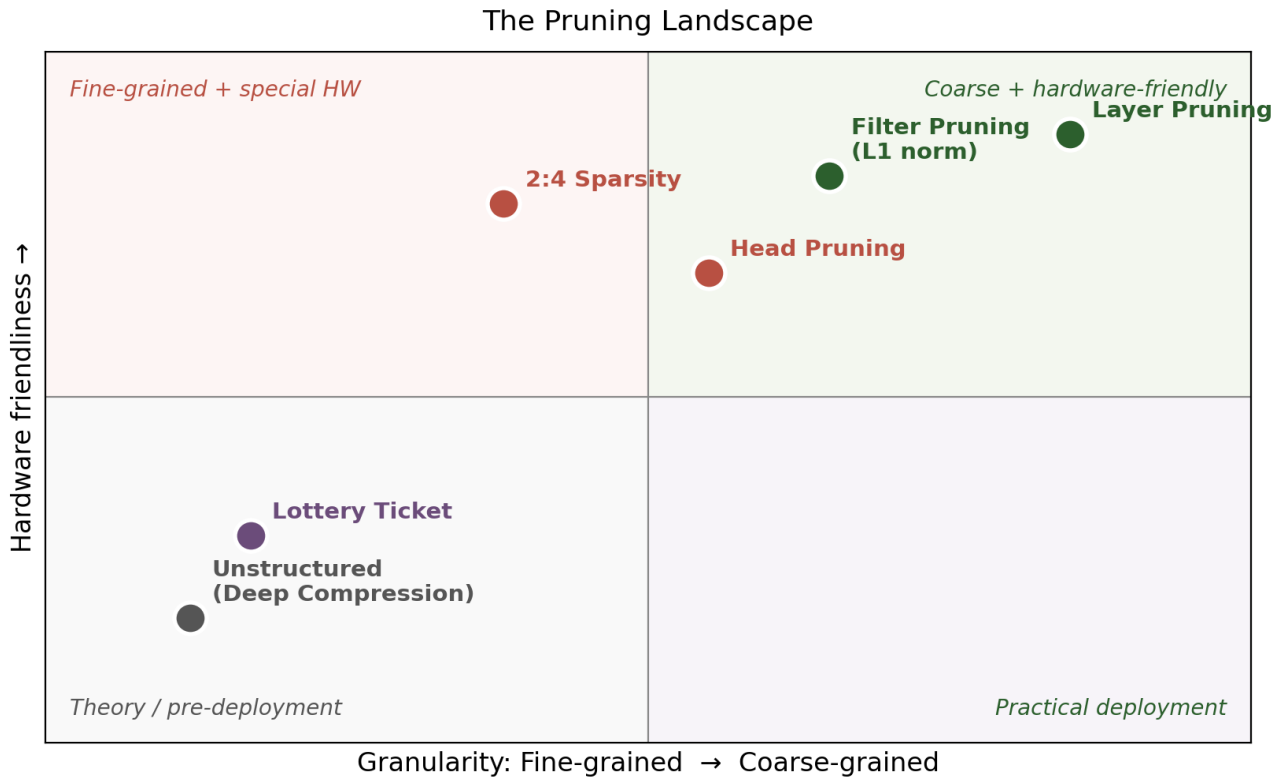
- Granularity: unstructured -> structured
- Hardware support: none -> specialized kernels

Key takeaways

- Pruning is architecture search in disguise
- Hardware dictates granularity choice
- Structured pruning wins in production

Bridges to other topics

- NAS (Week 5 ext), Efficient AI (Week 15)
- Combine with quantization, distillation for max gains



References

Key papers

- [1] Han et al. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. ICLR 2016
- [2] Li et al. Pruning Filters for Efficient ConvNets. ICLR 2017
- [3] Liu et al. Rethinking the Value of Network Pruning. ICLR 2019
- [4] Frankle & Carbin. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. ICLR 2019
- [5] Michel et al. Are Sixteen Heads Really Better than One? NeurIPS 2019
- [6] Frantar & Alistarh. SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot. ICML 2023
- [7] Sun et al. Wanda: A Simple and Effective Pruning Approach for LLMs. ICLR 2024

Further reading

- Hoefler et al. Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks. JMLR 2021 (survey)
- Han et al. Learning Both Weights and Connections. NeurIPS 2015