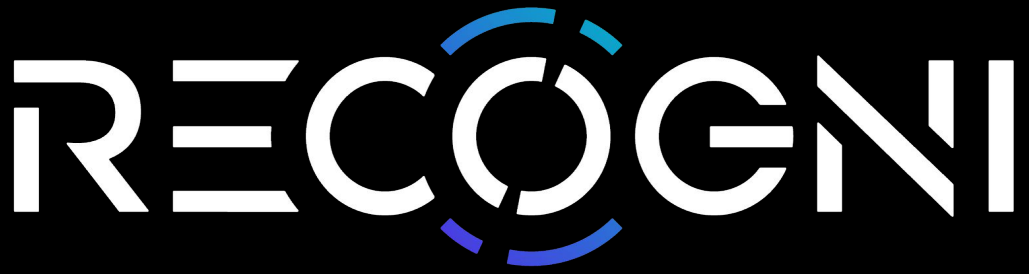

RECOGNI

Hardware-aware network compression: From data to silicon

Thomas Pfeil @
CODAI 2023

Abstract

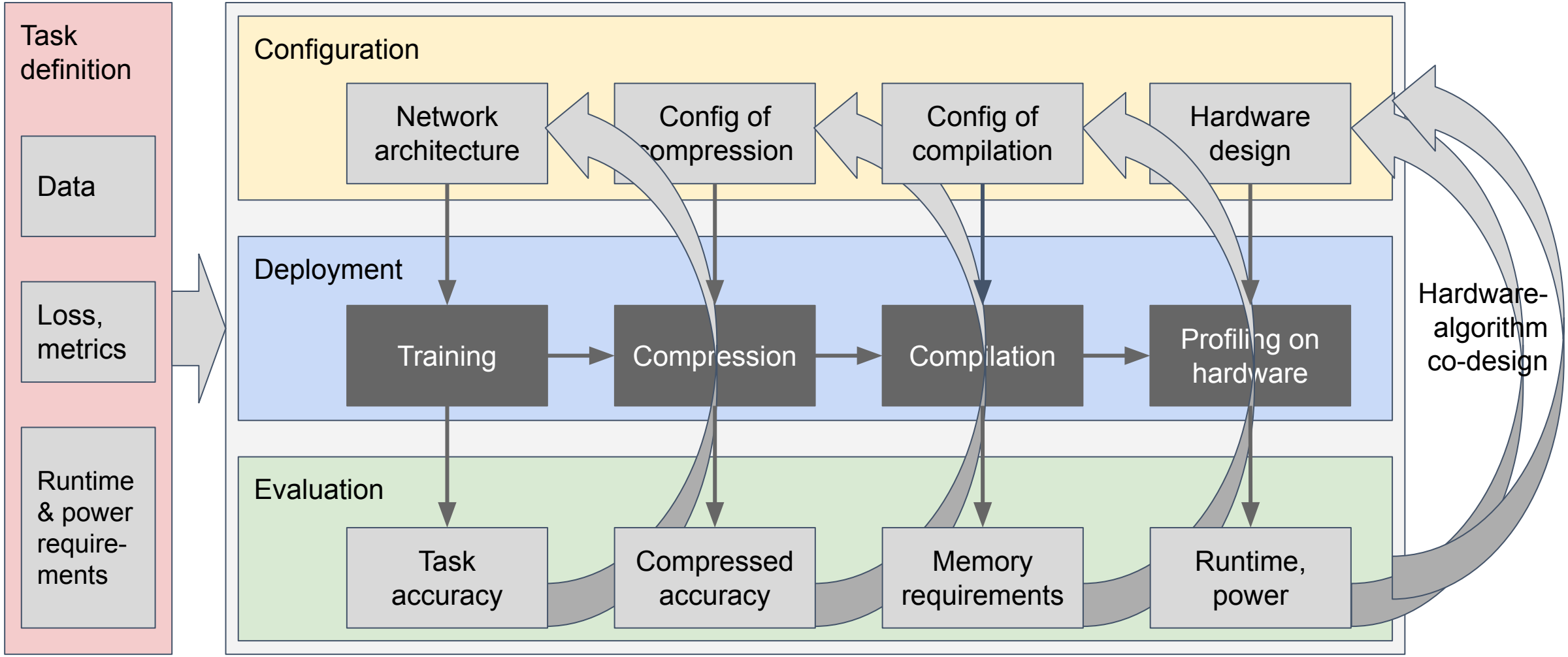
Deploying AI workload to hardware is a complex and challenging task, especially for specialized hardware systems that inherently support and exploit compression techniques. At a certain level of compression, optimizations at compiler level are not sufficient anymore to maintain task accuracy. To recover this task accuracy without sacrificing power efficiency, AI algorithms and hardware have to be co-designed. In this talk, I will present a holistic view on compression techniques for deep neural networks and their application in the context of specialized deep learning accelerators.

Outline

- Hardware-algorithm co-design
- QAT vs PTQ
- Second-Order Structured Pruning
- Conclusion

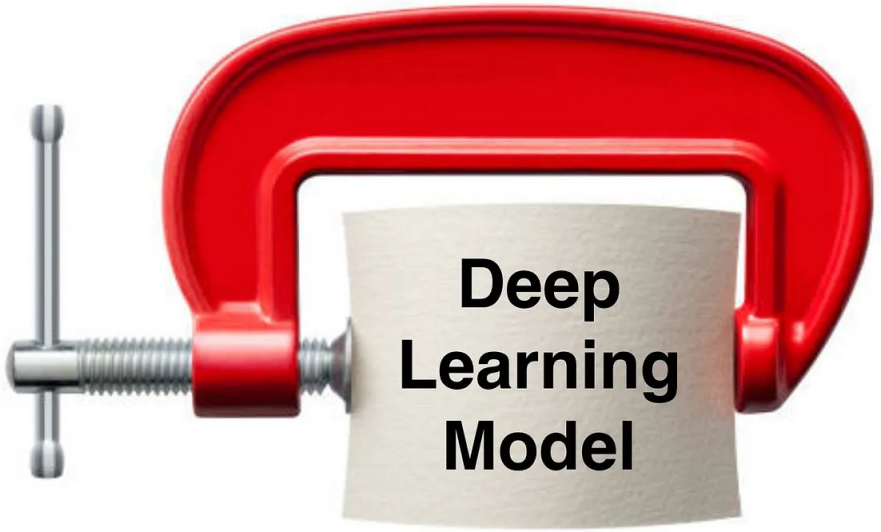
Hardware-algorithm co-design

Hardware-algorithm co-design



Compression techniques

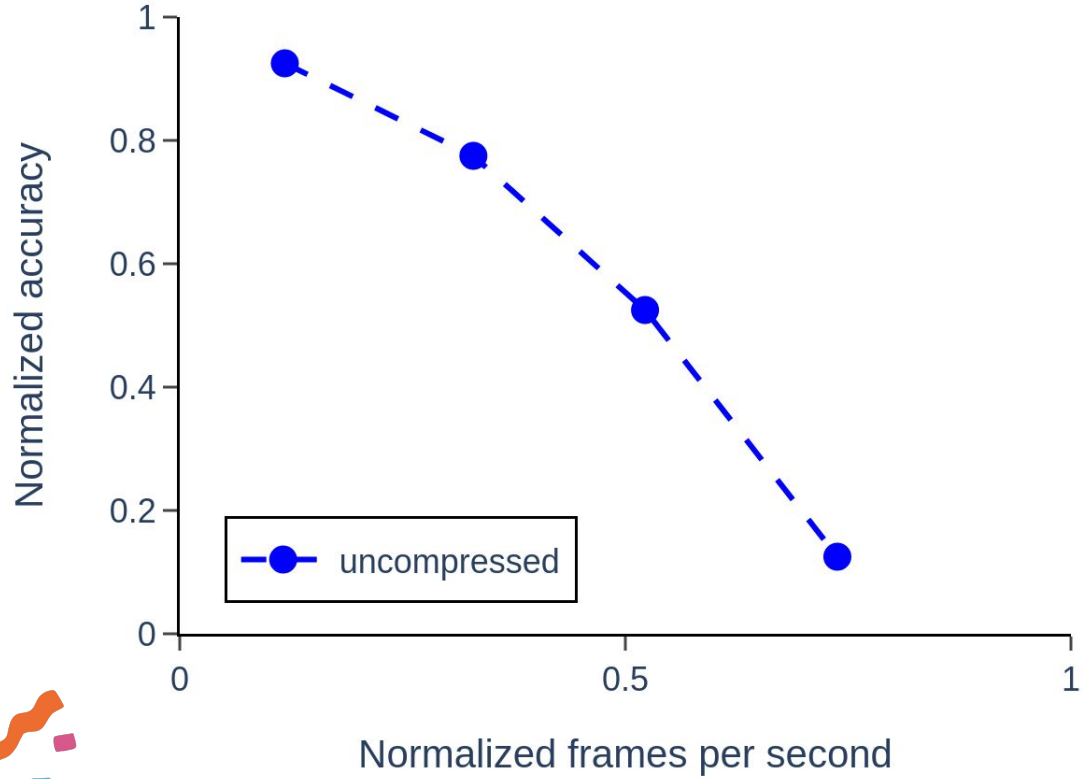
- Quantization
- Weight sharing
- Knowledge distillation
- Pruning
- Neural architecture search
  - Low-rank decomposition



Source: [1]

Hardware-algorithm co-design

- Accuracy and frames per second are of high interest for applications.
- Usually uncompressed networks are optimized for GPU execution and non-Pareto-optimal network architectures are discarded.



QAT vs PTQ

Quantization-Aware Training vs Post-Training Quantization

Method	Data	Network graph	Optimization method	Runtime	Task accuracy / Efficiency
	Interface to customer		Software		Application

Progressive Compression

Motivation:

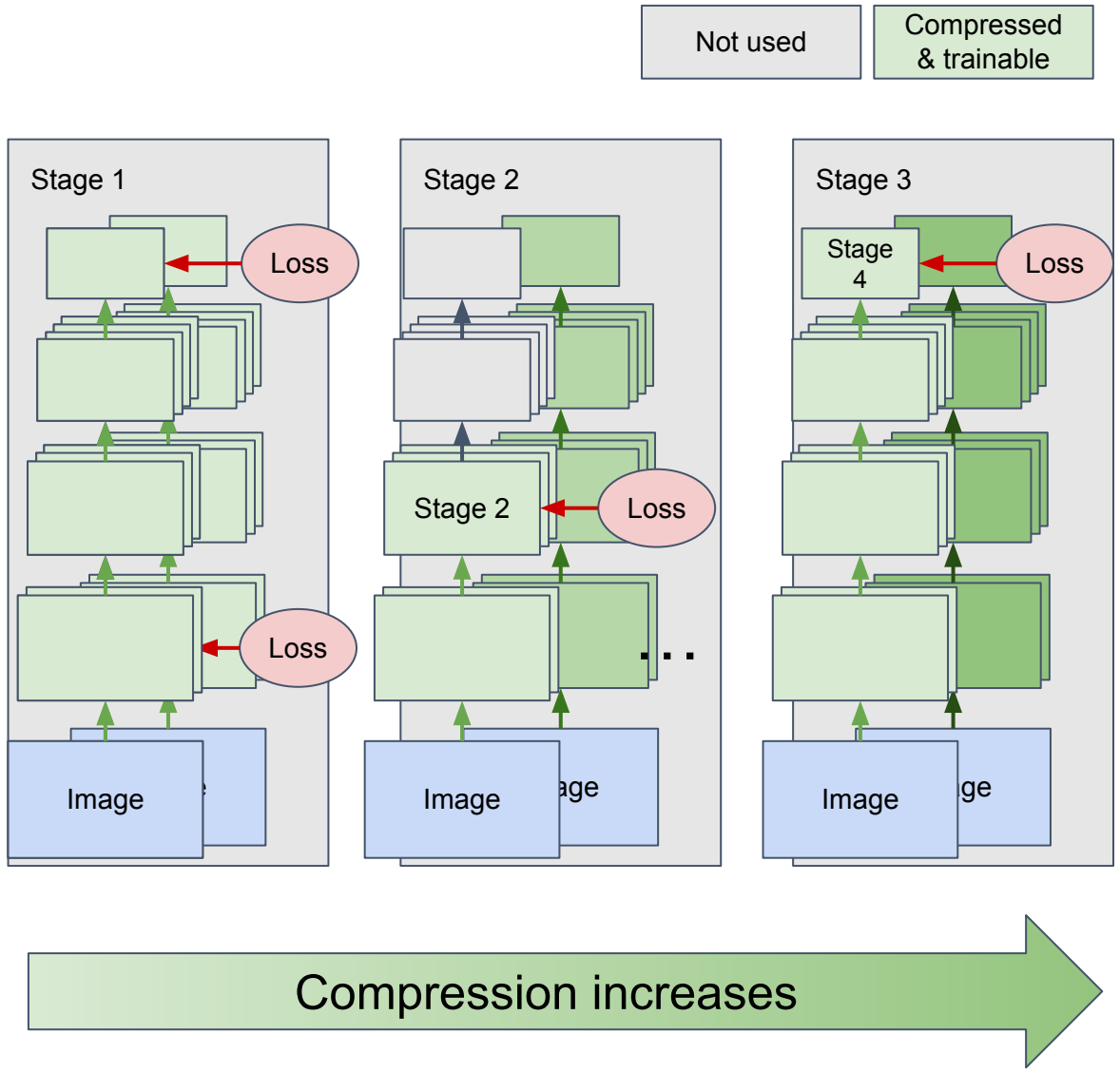
- Improve accuracy
- Reduce runtime

Hypothesis:

- The smaller the drops in accuracy between compression stages, the better the overall compression result.

Approach:

- Split the compression pipeline into stages that progressively increase compression.



Progressive Compression

Effective Training of Convolutional Neural Networks with Low-bitwidth Weights and Activations

Bohan Zhuang, Mingkui Tan, Jing Liu, Lingqiao Liu, Ian Reid, and Chunhua Shen

Progressive quantization [2]: 32-bit→8-bit→4-bit→2-bit

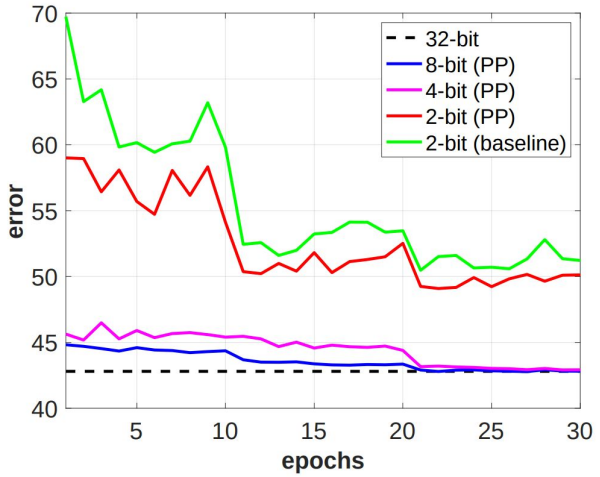


Fig. 3: The progressive training approach on AlexNet*.

Divide and Conquer: Leveraging Intermediate Feature Representations for Quantized Training of Neural Networks

Ahmed T. Elthakeb [1] **Prannoy Pilligundla** [2] **Fatemehsadat Mireshghallah** [2]
Alexander Cloninger [3] **Hadi Esmaeilzadeh** [2]

Progressive knowledge distillation [3]:

T: Trainable F: Freezed Q: Quantized H: High Precision

Bitwidth	Benchmark	AlexNet		ResNet-18		MobileNet-V2	
	Partitioning	3 Stages		3 Stages		3 Stages	
	Method	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
W32/A32	Full Precision	57.1	80.2	70.1	89.5	71.8	90.3
W4/A4	PACT	55.7	-	69.2	89.0	61.4	83.7
	LQ-Nets	-	-	69.3	88.8	-	-
	DSQ	-	-	69.6	-	64.8	-
	DoReFa	55.0	76.3	68.9	88.1	64.6	85.1
	DoReFa + DCQ	**56.2**	**79.2**	**69.9**	**89.2**	**66.2**	**87.3**
	Improvement	**0.89%**		**0.43%**		**2.47%**	

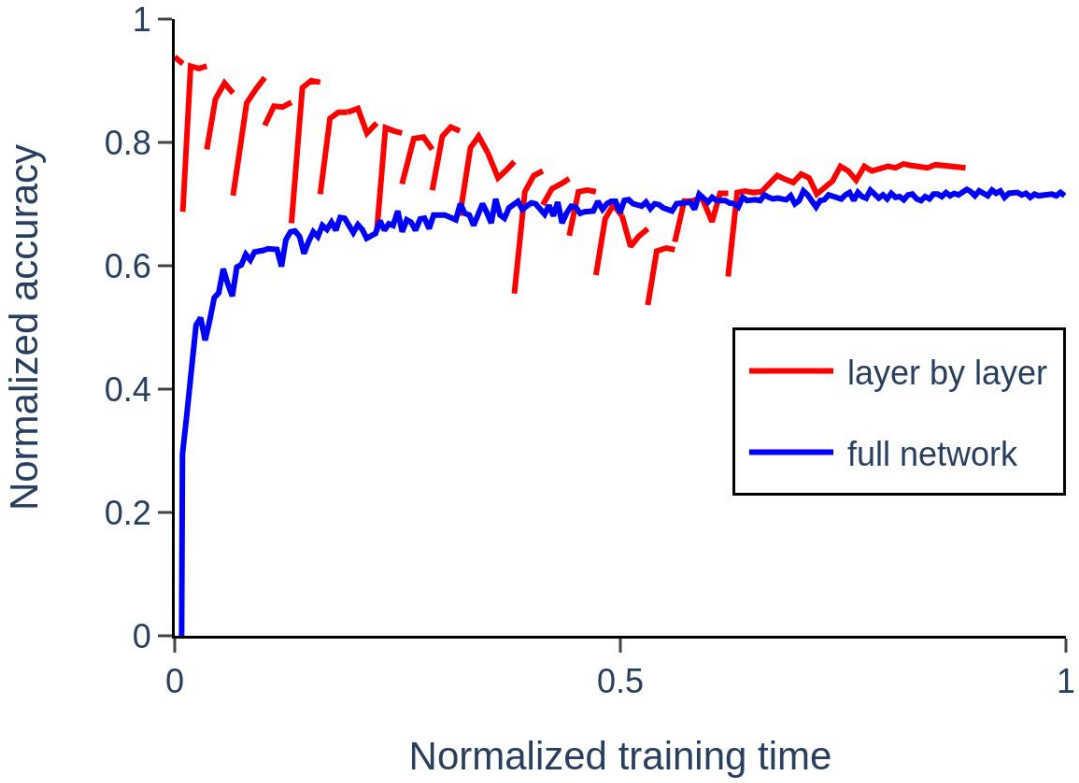
Progressive PTQ-KD

Methods:

- **Baseline:**
  Compress and fine-tune full network in a single stage.
- **Stage-by-stage approach:**
  Compress and fine-tune the network layer by layer.

Results:

- The drops in accuracy between stages is smaller for the stage-by-stage approach.
- Although the training time is shorter, a higher accuracy is achieved.
- To further decrease the runtime at the use of more compute resources stages can be optimized in parallel.



Second-Order Structured Pruning

SOSP

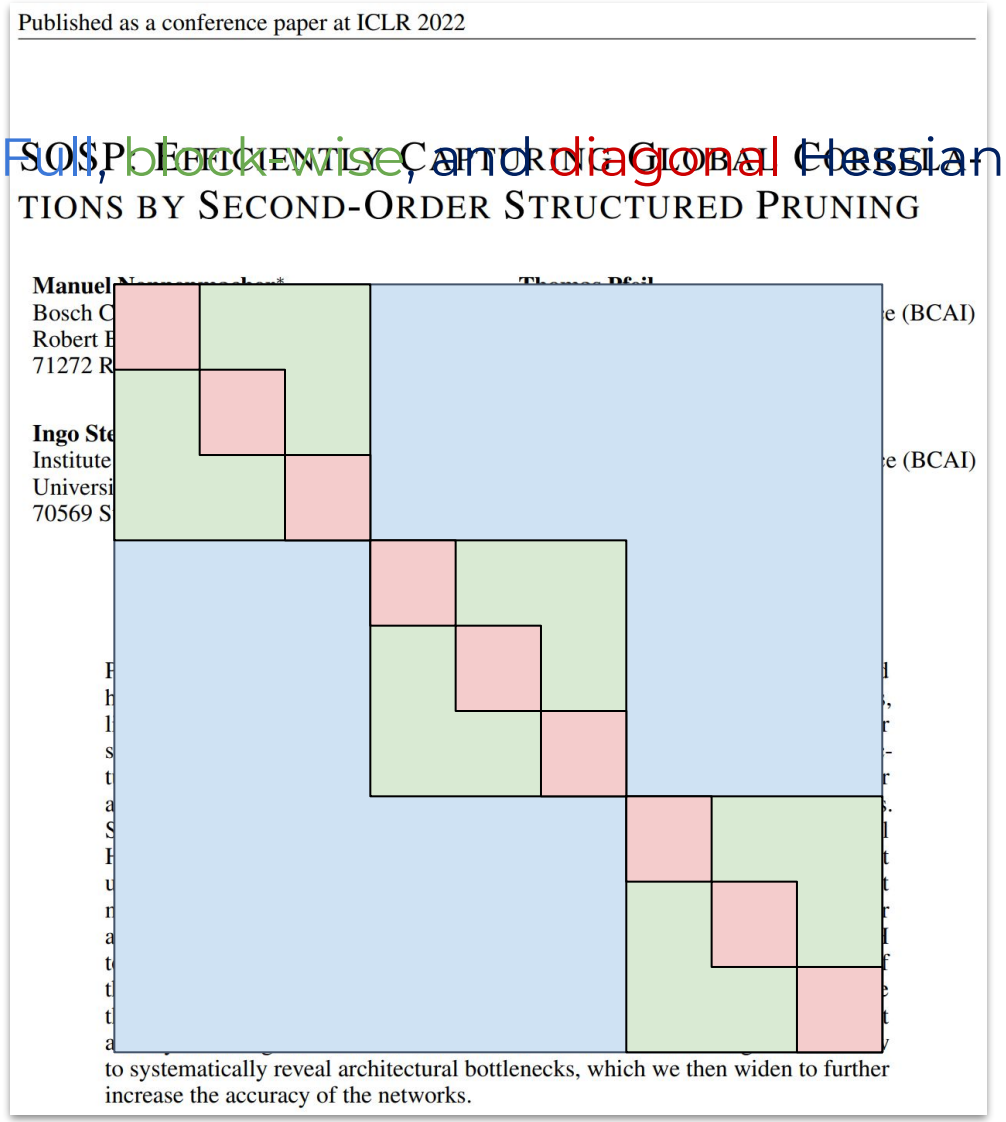
Second-Order Structured Pruning

Motivation:

- **One-shot** method
- using **second-order** correlations between structures
- allows for **global** optimization
- of **structured** pruning
- qualifying as efficient **NAS** method.

In contrast to:

- **Iterative** methods
- required because correlations are **not** considered
- only allow for **local** optimizations
- of oftentimes **unstructured** pruning
- **not** transferring to real-world applications.



Second-Order Structured Pruning - Methods

- **Objective:** “Select the pruning mask M to minimize the joint effect on the network loss.”
- The saliency of structures is computed by a **Taylor expansion** in which the second derivative captures correlation between structures.
- To reduce computational costs the Hessian-vector product is used to **approximate** the Hessian resulting in a complexity like for first-order methods.

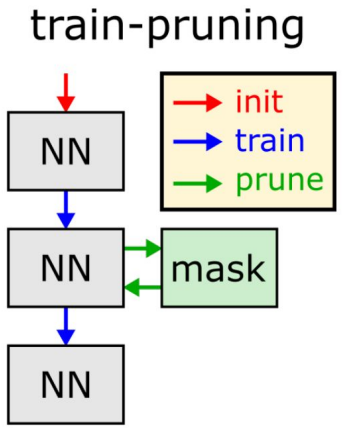
$$\lambda(M) := \left|\mathcal{L}(\theta) - \mathcal{L}(\theta_{\setminus M})\right|$$

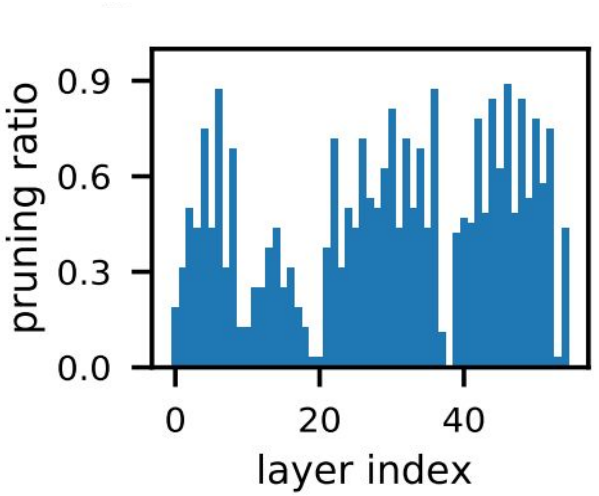
$$\lambda_2(M) = \left|\sum_{s\in M} \theta_s^T \frac{d\mathcal{L}(\theta)}{d\theta} - \frac{1}{2}\sum_{s,s'\in M} \theta_s^T \frac{d^2\mathcal{L}(\theta)}{d\theta\, d\theta^T}\theta_{s'}\right|$$

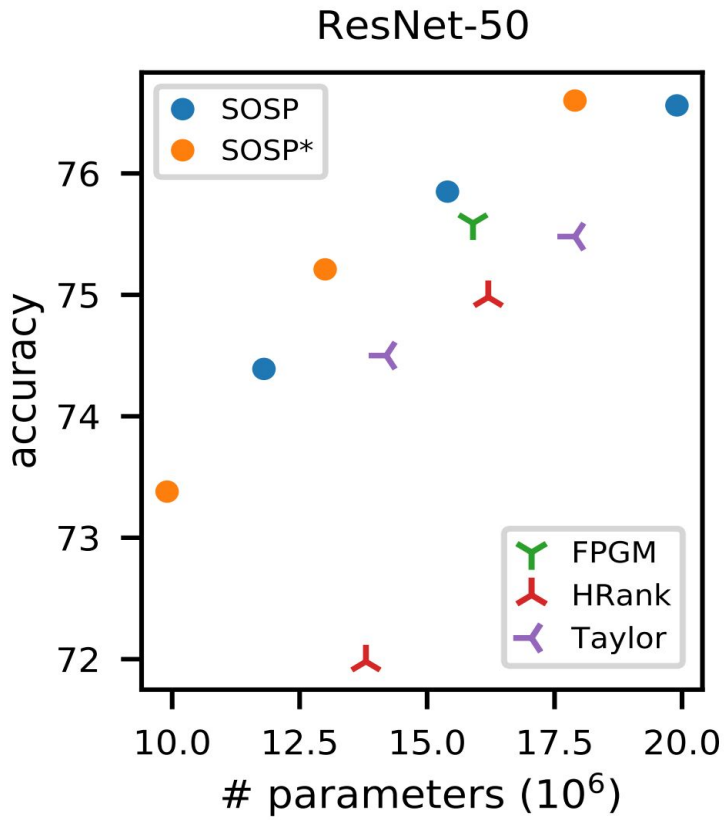
$$\lambda_2^H(s) := \left|\theta_s^T \frac{d\mathcal{L}(\theta)}{d\theta}\right| + \frac{1}{2}\left|\theta_s^T \left(H(\theta)\theta_{struc}\right)\right|$$

SOSP applied to ResNet-50 on ImageNet

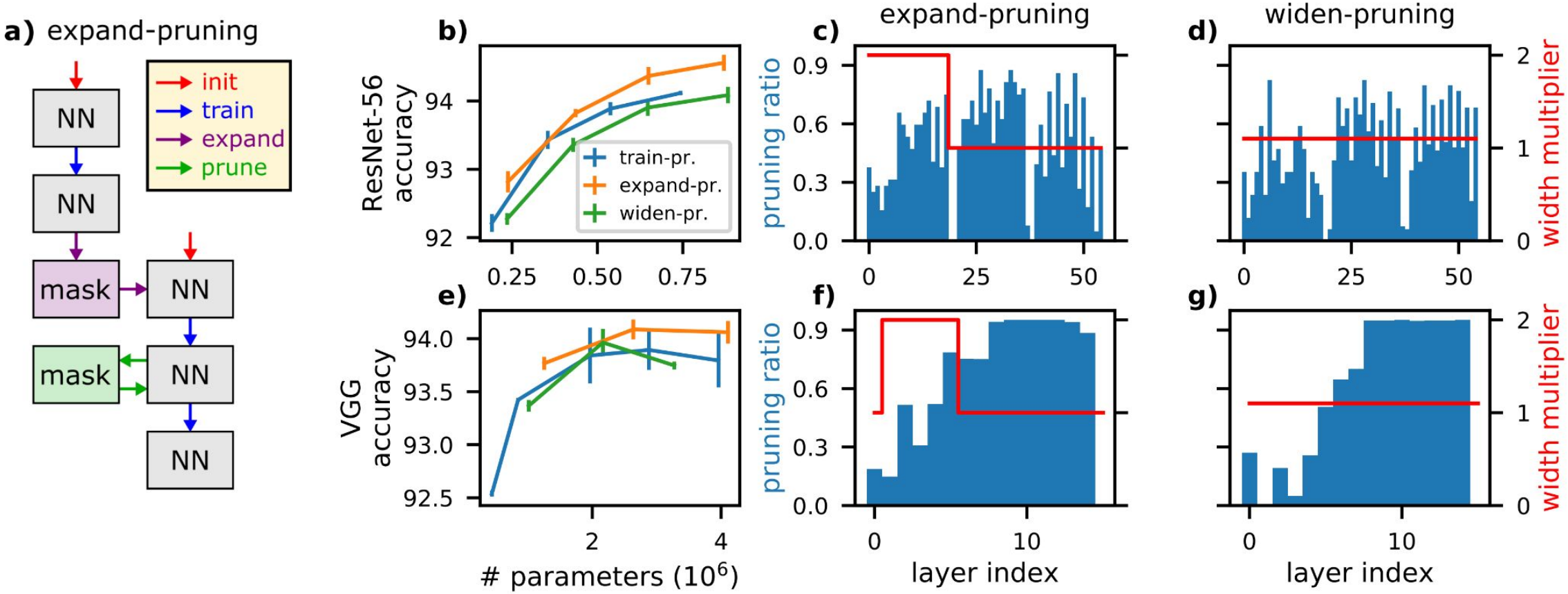
- Better trade-off between accuracy and number of parameters with less complex algorithm than comparable approaches.
- Works especially well for high pruning rates.







How SOSP can be used to find better neural architectures



Conclusion

- Closing the loop for **hardware-algorithm co-design** allows for rapid development of end-to-end optimized solutions.
- Under the assumption of limited access to data and the training pipeline, a **wide spectrum of PTQ methods** is available to compress neural networks. Methods on the Pareto-front of accuracy and FPS are of high interest.
- Effective methods exist for **global pruning** that qualify as a replacement or extension of neural architecture search methods.

Sources

- [1] https://medium.com/@wongsirikuln/cnn-model-compression-via-pruning-461c2fd167f6
- [2] Zhuang, Bohan, et al. "Effective training of convolutional neural networks with low-bitwidth weights and activations." IEEE Transactions on Pattern Analysis and Machine Intelligence 44.10 (2021): 6140-6152.
- [3] Elthakeb, Ahmed Taha, et al. "Divide and conquer: Leveraging intermediate feature representations for quantized training of neural networks." International Conference on Machine Learning. PMLR, 2020.
- [4] LeCun, Yann, John Denker, and Sara Solla. "Optimal brain damage." Advances in Neural Information Processing Systems 2 (1989).