U-Boost NAS: Utilization-Boosted Differentiable Neural Architecture Search

Ahmet Caner Yüzügüler Nikolaos Dimitriadis Pascal Frossard

EPFL, Switzerland

Motivation

- Inference latency is crucial in resource-constraint settings (e.g. mobile devices)
- ...but current DNN models underutilize resources
  - e.g., avg FLOPS/s utilization in Google's TPUv4 accelerator is 33%
- No prior work optimizes for hardware utilization
  - DNN models with low efficiency and increased latency in inference platforms
  - e.g. mismatches between DNN dimensions and target inference platforms are a major source of underutilization

Contributions

Hardware utilization analysis in DNN inference	⟶	sensitive to layer types, dimensions and can be improved while maintaining accuracy
Propose computational model for hardware util. in inference platforms	⟶	significantly higher accuracy vs prior models smooth relaxation enables differentiable NAS
Propose a differential NAS framework	⟶	optimize accuracy, inference latency, and resource utilization at inference platforms
Validation on CIFAR10, Imagenet100	⟶	significantly improve hardware utilization and inference latency on computer vision tasks

Resource Utilization in Inference Platforms

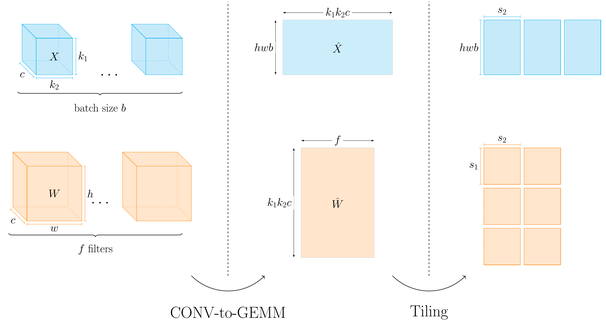
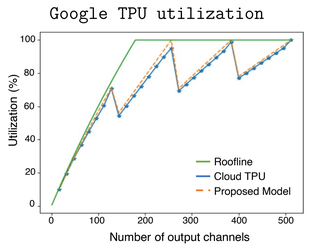


Figure 1. Mapping stages for CONV operations onto array-based architectures.

- Inaccurate HW design → discrepancies between estimated and actual runtimes
- Prior work only considers memory bottlenecks → simplistic and non-differentiable models
- We propose an analytical model to accurately approximate the behavior of target platforms.
- Our focus: specialized HW architectures such as systolic arrays (e.g. Google TPU or Tesla FSD)



$$\texttt{RUNTIME} = \left\lceil \frac{k_1 k_2 c}{s_1} \right\rceil \left\lceil \frac{f}{s_2} \right\rceil hwb$$

$$\texttt{UTIL} = \frac{k_1 k_2 c f}{s_1 s_2 \left\lceil \frac{k_1 k_2 c}{s_1} \right\rceil \left\lceil \frac{f}{s_2} \right\rceil}$$

Acknowledgments

The work of Ahmet Caner Yüzügüler was supported by the Hasler Foundation (Switzerland) and Nikolaos Dimitriadis was supported by Swisscom (Switzerland) AG.

Proposed NAS Framework

Smooth Approximation of ceiling function

Analytic utilization model is not differentiable → incompatible with Differentiable NAS. We use the generalised logistic function to obtain a smooth approximation of ceil function:

$$\texttt{CEIL}_{smooth}(x) = \sum_i \left[1 + \frac{\exp\left(-B(x - w_i)\right)}{C}\right]^{-1/v}$$

Multi-objective loss function

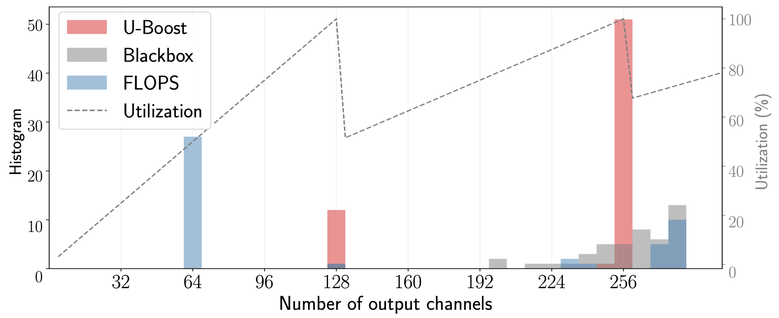
Let $\mathcal{F}$ be the hypothesis class of the search space and $\alpha \in \mathcal{F}$ the candidate architecture defining the function $f_\alpha : \mathcal{X} \to \mathcal{Y}$ for input and output domains $\mathcal{X}$ and $\mathcal{Y}$:

$$\mathcal{L}(\boldsymbol{x}, y, \alpha) = \mathcal{L}_{classification}(f_\alpha(\boldsymbol{x}), y) + \lambda \cdot \mathcal{L}_{latency}(\alpha) - \beta \cdot \mathcal{L}_{utilization}(\alpha)$$

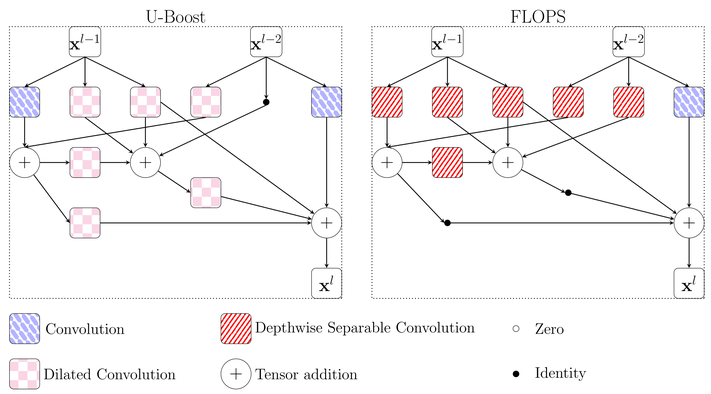
NAS algorithm: Hierarchical three-stage search

- *Microarchitecture*: layer types and connections with single-cell model (fixed channel dims)
- *Macroarchitecture*: optimal channel dims search cell-wise for model with k sequential cells
- *Final Training* of the selected architecture $\alpha \in \mathcal{F}$

Qualitative Perspective on the algorithm



- U-Boost selects only the channel dimensions that achieve full utilization
- The baselines are unable to model the search space correctly and result in underutilization



- DWS Convolutions are deemed as important for low runtime but they result in underutilization
- FLOPS and other baselines select DWS Convolutions (as low as 1% utilization)
- However, U-Boost avoids DWS and instead selects (Dilated) convolutions

Experiments

- NAS experiments with different computational model for utilization
- Measured top-1 test accuracy, runtime by our custom cycle-accurate HW simulator, utilization
- Baselines:
  - **FLOPS**: assumes full utilization, latency=#operations divided by the theoretical peak throughput
  - **Roofline**: compute-bound→ same as the FLOPS, memory-bound mode→ latency=memory footprint size divided by off-chip bandwidth
  - **Blackbox**: lookup table storing latency values for all layer types and dimensions

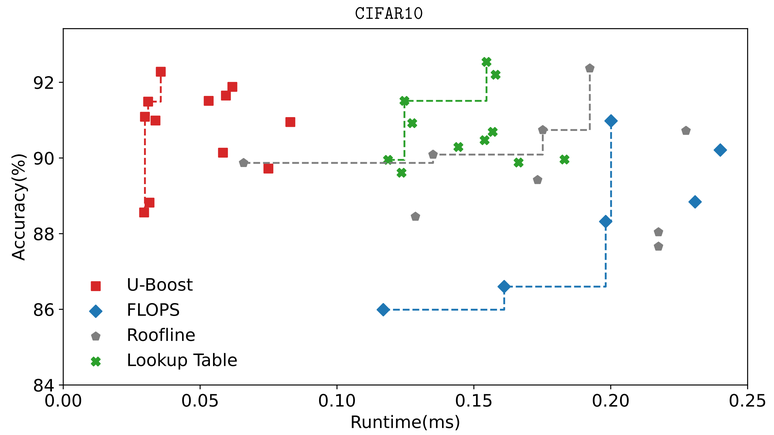


Figure 2. Experiments on CIFAR10 dataset. Upper left corner is optimal. The dashed lines connect the points in the Pareto Front of each method. The points are produced for different values of the latency regularization coefficient λ.

- *Accuracy*: All method achieve comparable results
- *Runtime*: $2.8 - 4\times$ speedup
- Incorporating Utilization to NAS objectives improves runtimes *without* accuracy deterioration

ImageNet100

	Acc. (%, ↑)		Runt. (ms, ↓)		Util. (%, ↑)		HV (↓)	# Params	
λ	0.1	1.0	0.1	1.0	0.1	1.0	(across λ)	0.1	1.0
Blackbox [3]	87.5	87.8	4.8	4.05	69.3	68.5	49.4	70.5	55.5
Roofline [2]	86.5	84.0	4.7	3.5	6.8	4.8	72.2	13.7	5.7
FLOPS [1]	87.2	78.4	6.1	3.45	5.5	3.1	108	14.4	3.5
U-Boost	87.8	87.9	2.2	1.05	91.1	78.6	**12.7**	47.3	30.1

- FLOPS and Roofline result in severe underutilization ($< 10\%$) and high runtimes
- Blackbox opts for the largest possible network leading to better utilization but poor runtimes
- U-Boost achieves the optimal combination for all metrics: Without losing in accuracy, Harware utilization is modeled correctly and runtime is accordingly low!

Conclusion

- Showed the importance of resource utilization in runtime characteristics on inference platforms
- Proposed U-Boost, a utilization-aware differentiable NAS method: incorporating analytical resource utilization model allows for efficient and accurate estimation
- U-Boost NAS achieves $2.8 - 4\times$ inference latency speedup with similar or improved accuracy

References

[1] Ariel Gordon et al. "MorphNet: Fast & Simple Resource-Constrained Structure Learning of Deep Networks". In: *CVPR 2018*. 2018.
[2] Sheng Li et al. "Searching for Fast Model Families on Datacenter Accelerators". In: *CVPR 2021*.
[3] Bichen Wu et al. "FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search". In: *CVPR 2019*.