MoGA: Searching Beyond MobileNetV3

Xiangxiang Chu, Bo Zhang and Ruijun Xu

AI Lab, Xiaomi Inc.

Objectives

1. Mobile-GPU awareness in neural architecture search
2. Multi-objective solution to best profit "once-for-all" feature
3. Reduced adaptation cost $\boldsymbol{O(1)}$ for new hardware

Introduction

- In this paper, we aim to bring forward the frontier of mobile neural architecture design by utilizing the latest neural architecture search (NAS) approaches. First, we shift the search trend from mobile CPUs to mobile GPUs, with which we can gauge the speed of a model more accurately and provide a production-ready solution. On this account, our overall search approach is named Mobile GPU-Aware neural architecture search (MoGA). Second, we replace traditional multi-objective optimization with a weighted fitness strategy where we lay more attention on accuracy and latency, other than the number of parameters. Third, we benefit from the recent one-shot supernet training [1] and build an accurate latency look-up table. The overall NAS pipeline costs **12 GPU days**, about **200**$\times$ less than MnasNet. Finally, we present our searched architectures that outperform MobileNetV3. Namely, MoGA-A achieves an outstanding **75.9%** top-1 accuracy on ImageNet, MoGA-B 75.5% and MoGA-C 75.3%. Remarkably, MoGA-A achieves **0.9%** higher accuracy than MobileNetV3 with only 1***ms*** increased latency on mobile GPUs.

Motivations

- Mobile GPU latencies are not linear to CPU.
  - Current design is mobile CPU-based but deployed onto GPUs in real applications.
- Avoid overfitting: more parameters and less multiply-adds are better.
  - Hint from the MobileNet trilogy.
- Neural architecture search needs hardware-awareness.
  - Prove hardware-aware NAS generates different architectures w.r.t chips.
  - Provide industry-level production-ready models.

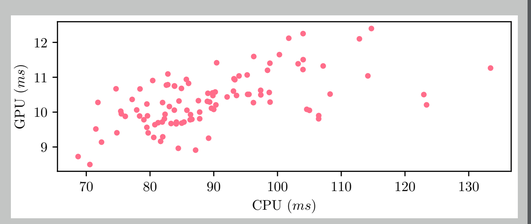

Figure 1:Latency relationship on mobile CPUs vs. on mobile GPUs.

Methods

- Stage 1: "once-for-all" one-shot supernet training
  - Search space: ProxylessNAS [2] adapted for MobileNetV3-Large [3]
  - Train the supernet with Strict Fairness strategy [1]
- Stage 2: Weighted multi-objective search
  - a well-known evolutionary method: NSGA-II
  - weighted crowding distance to give importance to different objectives
  - hierarchical mutation from MoreMNAS
  - a latency lookup table (device-specific, based on basic operators)
- Ablation Study
  - Two objective (worse) vs. Three (better)
  - Random mutation (worse) vs. Hierarchical mutation (better)

Mathematical Section

- Problem Formulation

$$\begin{aligned} &\text{minimize } \{-\boldsymbol{Acc(m)}, \boldsymbol{Lat(m)}, -\boldsymbol{Params(m)}\}, \forall \boldsymbol{m} \in \boldsymbol{\Omega} \\ &\boldsymbol{s.t.} \quad \boldsymbol{w_{acc} + w_{lat} + w_{params} = 1}, \forall \boldsymbol{w} >= \boldsymbol{0} \end{aligned} \tag{1}$$

  - For practical applications, we set $\boldsymbol{w_{acc} = w_{lat} = 0.4}, \boldsymbol{w_{params} = 0.2}$ in our experiment.
- Weighted crowding distance for NSGA-II

$$\boldsymbol{D(m_j) = \sum_{i=1}^{n} w_i * \frac{O^i_{neighbor+} - O^i_{neighbor-}}{O^i_{max} - O^i_{min}}}. \tag{2}$$

Results: Table

- Comparison of SOTA mobile models

Methods	$\times+$ (M)	$\boldsymbol{P}$ (M)	$\boldsymbol{L_g^S}$ (ms)	$\boldsymbol{L_g^M}$ (ms)	$\boldsymbol{L_c}$ (ms)	Top-1 (%)
MobileNetV2	300	3.4	6.9†	7.0†	78	72.0
MobileNetV3 [3]	219	5.4	10.8*	9.5*	66	75.0*
MnasNet -A1	312	3.9	-	-	78	75.2
MnasNet - A2	340	4.8	-	-	84	75.6
FBNet-B	295	4.5	-	-	23‡	74.1
Proxyless-R	320†	4.0	7.3†	7.9†	78	74.6
Proxyless GPU	465†	7.1	9.6†	9.8†	124	75.1
Single-Path	365	4.3	-	-	79	75.0
Once for All	327	-	-	-	112*	75.3
FairNAS-A [1]	388	4.6	9.8†	9.7†	104	75.3
MoGA-A (Ours)	304	5.1	11.8	11.1	101	**75.9**
MoGA-B (Ours)	248	5.5	10.3	10.0	81	75.5
MoGA-C (Ours)	221	5.4	9.6	8.8	71	75.3

Table 1:Comparison of mobile models on ImageNet. $\boldsymbol{P}$: Number of parameters, $\boldsymbol{L_g^S}$ ($\boldsymbol{L_g^M}$): SNPE (MACE) latency on mobile GPU, $\boldsymbol{L_c}$: TFLite latency on CPU *: Our reimplementation. †: Based on its published code. ‡: Samsung Galaxy S8. *: Samsung Note 8.

Results: Figure

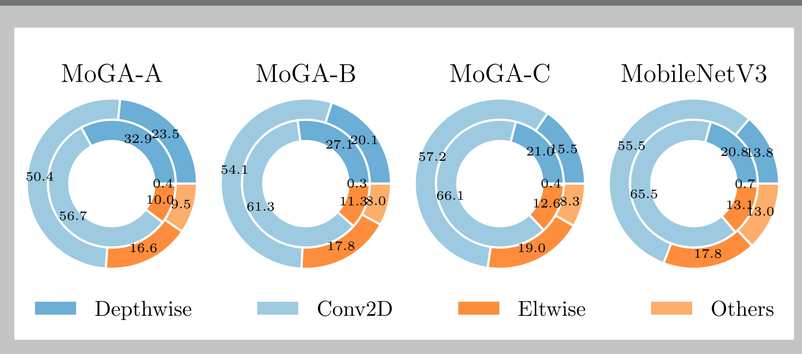

Figure 2:Latency pie chart of MoGA-A, B, C and MobileNetV3 operations when run on mobile CPUs (inner circle with TFLite) vs. on mobile GPUs (outer circle with MACE).

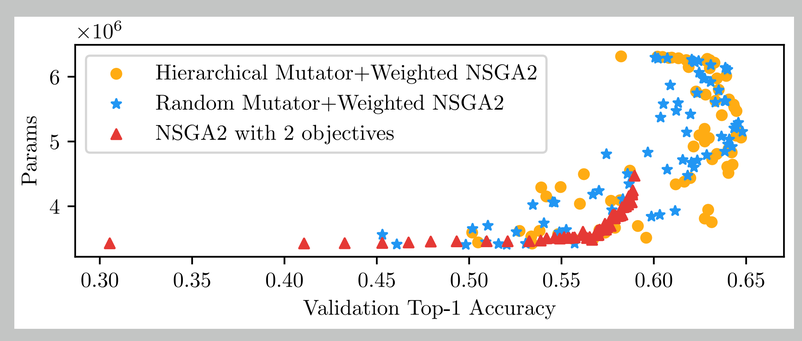

Figure 3:Pareto Front of weighted NSGA-II with hierarchical mutator compared with that of a random mutator and of two objectives (accuracy, latency).

Conclusion

- first Mobile GPU-Aware (MoGA) solution
- weighted fitness strategy to comfort more on latency and accuracy than parameters
- reduced search cost (12 GPU days, 200$\times$ less than MnasNet)
- new SOTA results surpassing MobileNetV3

References

[1] Xiangxiang Chu, Bo Zhang, Ruijun Xu, and Jixiang Li.
FairNAS: Rethinking Evaluation Fairness of Weight Sharing Neural Architecture Search.
arXiv preprint. arXiv:1907.01845, 2019.

[2] Han Cai, Ligeng Zhu, and Song Han.
ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware.
In *International Conference on Learning Representations*, 2019.

[3] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al.
Searching for MobileNetV3.
In *International Conference on Computer Vision*, 2019.

Acknowledgments

- We thank Xiaomi MACE team for support on mobile latency measurements.