Neural architecture search with target hardware control

Firsov Sergey
Intelligent Systems Phystech

December 21, 2024

Abstract

The paper examines the problem of deep learning model selection with respect to target hardware. We propose a method for neural architecture search (NAS) that accounts for hardware constraints and model complexity. The complexity of the model is measured by the number of parameters, while the hardware constraints are represented by the overall latency of network operations on the target device.

This approach builds upon Differentiable Architecture Search (DARTS), treating network structural parameters as functions of a complexity parameter, and extends it by incorporating latency-aware optimization inspired by FBNet. By introducing Gumbel-Softmax sampling and hypernetworks, we enable simultaneous optimization of multiple architectures across a range of complexity levels. This results in a single optimization process that identifies a family of architectures tailored to different computational budgets, reducing search time and resources.

1 Introduction

Selecting an appropriate architecture for deep learning models is a crucial task that directly impacts model efficiency and performance. With deep learning continuing to push computational limits, researchers face the challenge of finding a balance between model complexity, accuracy, and resource consumption. Recent advances in Neural Architecture Search (NAS) [2] techniques, such as Differentiable Architecture Search (DARTS) [3], seek to automate this process by exploring large search spaces of possible network structures. However, these methods often struggle with high computational requirements and the need for architecture adjustments when model complexity or target hardware changes [5].

One of the significant developments in NAS is the introduction of hardware-aware models. For example, FBNet [4] incorporates latency into the architecture search process, optimizing not only model performance but also hardware efficiency. This approach addresses the mismatch between FLOPs and actual hardware performance, a limitation of many prior NAS methods. FBNet achieves this through gradient-based optimization and Gumbel-Softmax sampling, which dramatically reduce the search costs while generating a family of hardware-optimized models.

Similar ideas are used in ProxylessNAS [1], which solves the problem of high memory and computing costs by optimizing architectures directly on large tasks and target hardware platforms, without using proxy tasks. ProxylessNAS introduces a direct search engine (Reinforce sampling) for learning architectures on large datasets and simulates operation delays to account for hardware limitations. However, this approach requires careful calibration, as enhanced learning often suffers from high variance.

Building on these ideas, our work improves upon DARTS-CC [5] , a NAS approach that uses hypernetworks to control model complexity during architecture search. Unlike other methods that search for individual architectures at different complexity levels, DARTS-CC generates multiple architectures in a single optimization process. Inspired by FBNet, we extend DARTS-CC by integrating latency-aware optimization and replacing the scalar complexity parameter with a simplex-based representation of architecture choices. This enables simultaneous search for architectures optimized across multiple complexity and latency levels, further reducing NAS time, and ensuring deployability across diverse hardware environments.

Our contributions can be summarized as follows:

- We extend DARTS-CC by incorporating hardware latency into the optimization process. This enables the discovery of architectures that are not only accurate, but also efficient on target devices.

- We replace the scalar complexity parameter in DARTS-CC with a simplex-based representation, allowing the use of Gumbel-Softmax sampling for flexible architecture optimization.
- We propose a unified framework that combines latency-aware optimization, hypernetworks, and Gumbel-Softmax sampling to generate a family of architectures in a single optimization run.
- We validate our method on multiple datasets and hardware platforms, demonstrating improved efficiency and flexibility compared to existing NAS approaches.

1.1 Proposed Method

The proposed method formulates Neural Architecture Search (NAS) as an optimization problem that simultaneously considers model accuracy, complexity, and hardware efficiency.

Problem Formulation. In conventional DARTS the goal is to find architectural parameters $\boldsymbol{\alpha} \in \mathcal{A}$ and model weights $\boldsymbol{w} \in \mathcal{W}$ such that the loss function $L(\boldsymbol{w}, \boldsymbol{\alpha})$ is minimized. The optimization problem can be formulated as follows:

$$\min_{\boldsymbol{\alpha} \in \mathcal{A}} \big[L(\boldsymbol{w}^*(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \big],$$

subject to:

$$\boldsymbol{w}^*(\boldsymbol{\alpha}) = \arg\min_{\boldsymbol{w} \in \mathcal{W}} L(\boldsymbol{w}, \boldsymbol{\alpha}),$$

where $L(\boldsymbol{w}, \boldsymbol{\alpha})$ represents the task loss.
However, DARTS does not account for flexibility in controlling model complexity or hardware constraints. So we propose these extensions to the DARTS framework to address its limitations:

First Step. Architecture Representation. The architectural parameters $\boldsymbol{\alpha}$ we define as a function of a complexity parameter $\lambda \in \mathbb{R}^+$ [5], which maps to the architecture space $\mathcal{A}$:

$$\boldsymbol{\alpha} : \lambda \to \mathcal{A}.$$

This allows generates multiple architectures with different complexity in a single optimization process.

Second Step. Simplex-Based Complexity Control. Instead of using a scalar parameter λ, we introduce a simplex-based representation $\boldsymbol{S} \in \Delta^{k-1}$, where Δ^{k-1} denotes a $(k-1)$-dimensional simplex, k - is number of operations. The parameter $\boldsymbol{S}$ is sampled using the Gumbel-Softmax distribution to allow differentiable selection of architectural components:

$$\boldsymbol{S} \sim \text{GumbelSoftmax}(\boldsymbol{\theta}).$$

Third Step. Latency regularization. To account for hardware constraints, we introduce a latency lookup table that records the latency of each operation on the target device. The total latency is computed as a weighted sum of the latencies of the operations selected by the architecture.

The objective function now becomes:

$$\mathbb{E}_{\boldsymbol{S} \sim \text{GumbelSoftmax}} \big[L(\boldsymbol{w}^*(\boldsymbol{S}), \boldsymbol{\alpha}(\boldsymbol{S})) + \lambda \cdot \text{Reg}(\boldsymbol{\alpha}(\boldsymbol{S})) + \kappa \cdot \text{Latency}(\boldsymbol{\alpha}(\boldsymbol{S})) \big].$$

So by using complexity dependence architecture parameters we generate multiple architectures in a single optimization process, by replacing the scalar complexity parameter with the simplex representation, we achieve finer control over architectural complexity, enabling flexible optimization across a range of hardware and resource constraints and by using latency lookup table, we can control hardware constraints.

References

[1] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware, 2019. https://arxiv.org/abs/1812.00332.

[2] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21, 2019.

[3] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2019.

[4] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[5] Konstantin Yakovlev, Olga Grebenkova, Oleg Bakhteev, and Vadim Strijov. Neural architecture search with structure complexity control. In *Recent Trends in Analysis of Images, Social Networks and Texts*, pages 207–219, 2022.