Discovering Efficient Neural Architecture for Resource-Constrained Devices with Explicit Constraints

Kwan Hee Lee, Chan Yeong Park, Hyang-Won Lee
Dept. of Computer Science Engineering, Konkuk Univ. Seoul.
{kwan7595, atoz89961, leehw}@konkuk.ac.kr

Abstract

The performance of deep learning(DL) heavily depends on computing resources, and hence, it is hard to deploy DL-based services at end devices with limited resources. To overcome this limitation, the search for lightweight architectures has been extensively studied with various techniques. In this paper, we propose a novel neural architecture search method for meeting explicit resource constraints without significant loss of accuracy.

Ⅰ. Introduction

The development of artificial intelligence(AI) led to a demand for new, better architectures. On the other hand, many tasks related to Artificial Intelligence are deployed onto a variety of devices with limited computing resources. To serve tasks with best-performing neural architectures, the resource constraints of each device make it hard to deploy. Classical state-of-the-art(SOTA) architectures are designed manually by human experts, but making architectures for every different device is very hard and expensive.

To find a novel architecture without designing, neural architecture search(NAS) comes into hand. We refer readers to [1], for more details. Different kinds of searching algorithms have been developed to search for the best architectures(Bayesian optimization[2], evolutionary algorithm[3],reinforcement learning[4]), but suffered from vast amount of computational power(800 GPU days). gradient-based methods[5] emerged for faster searching, making it easier to search for architectures and parameters in one-shot method, but still suffering from sub-optimality and memory issues.

To overcome limitations on computing powers, network pruning comes into the limelight. The goal of pruning is to downsize the neural network's size for easier deployment on mobile devices with resource constraints. Triggered by[6], a network can be compressed up to 90% while not compromising accuracy. But sparse architectures generated by the pruned network are hard to train from scratch.

To break these limitations, we combine network pruning and gradient-based searching algorithm by formulating two objectives with bilevel optimization to search for the best architectures that meet resource constraints. With the CIFAR10 dataset, we show that our method can compress architecture up to 89% of full-size architecture while losing only 2% of accuracy.

Ⅱ. Method

Graph Structure : We formulate neural graph as $G = (V, E)$, consisting of nodes V and edges $E \sqsubseteq V \times V$. At each node, we apply weighted summation of incoming node and edges, each edge assigned with a weight w_{uv}. and we perform activation-normalization, 3x3 depth-wise separable convolution with the given input. this formulation can be seen as a 3x3 convolution at each node with an information flow on a complete bipartite graph. To discover neural architectures within this formulation, we consider edges that are chosen as real edge E, and counterparts as *hallucinated edge* $\mathrm{E}_{\mathrm{hal}}$. main concept of this algorithm is to update ***hallucinated edges*** at the backpropagation step, meaning the set of real edges can be changed while training. we follow the same formulation of the static neural graph at [7], we refer readers for more details.

Problem Formulation: In this paper, to discover neural wirings while satisfying performance constraints, we formulate our problem as follows.

$$\min_{w} L_{param,Val}(w, \theta^*),$$
$$s.t.\ \theta^* = \min_{\theta} L_{CE,Train}(w, \theta)$$

we replace top-k constraint with weight threshold to solve our problem, and formulated as bilevel optimization. in train step, update edge weight w and model parameter θ subject to cross entropy loss, and at validation step we update edge weight with given T, subject to parameter loss to search for better neural wiring.

Loss for Resource Constraints: To satisfy resource constraints, we design a loss function L_{param}.

$$L_{param} = \beta * log\left(1 + e^{(F(w)-T)}\right)$$
$$F(w) = expected\ \ parameter$$

(β = scale parameter, T = target parameter) parameter loss forces edge weights closer to the threshold with gradient update if constraints are violated, to satisfy target parameters. Otherwise, only CE Loss takes action to search for architecture with given target parameters. The expected parameter of architecture is continuously relaxed with probability to make parameter loss differentiable. By using both loss functions with bilevel optimization formulation, our goal is to search for the best architectures that satisfy performance constraints.

Ⅲ. Experimental Results

We used CIFAR10 dataset, and made a small classifier(about 30M at full-size architecture) to show that our algorithm can find efficient neural architecture without compromising accuracy. Target # parameters were 20M~ to 10M, and fig1. shows achieved # parameters/accuracy

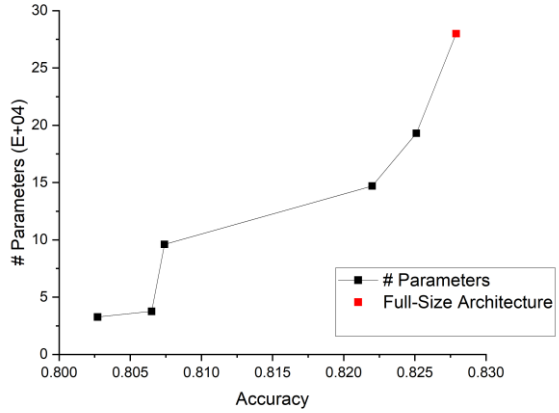

Fig 1. The number of parameters and accuracy with different performance constraints

As shown in fig1, we can compress architecture size up to 89% of full-size architecture, while losing only 2% accuracy.

IV. Conclusion

In this paper, we present a method to find efficient neural architecture of performance constraints given as the number of parameters. The main concept is to use both loss functions, formulated as a bilevel optimization problem, to search for efficient neural wiring with given learnable parameters under performance constraints. This work suggests a method to deploy efficient neural networks for a variety of tasks, where devices have limited computing power.

ACKNOWLEDGMENT

This work was supported in part by the Human Resources Program in Energy Technology of the Korea Institute of Energy Technology Evaluation and Planning (KE-TEP) and the Ministry of Trade, Industry ₩& Energy (MO-TIE) of the Republic of Korea (No.20204010600220), and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2021R1A2C2012801).

REFERENCES

[1] Elsken, T., J. H. Metzen, and F. Hutter. "Neural architecture search: A survey. arXiv 2018." *arXiv preprint arXiv:1808.05377* (2018).

[2] Baker, Bowen, et al. "Designing neural network architectures using reinforcement learning." *arXiv preprint arXiv:1611.02167* (2016).

[3] Real, Esteban, et al. "Large-scale evolution of image classifiers." *International Conference on Machine Learning*. PMLR, 2017.

[4] Zoph, Barret, and Quoc V. Le. "Neural architecture search with reinforcement learning." *arXiv preprint arXiv:1611.01578* (2016).

[5] Liu, Hanxiao, Karen Simonyan, and Yiming Yang. "Darts: Differentiable architecture search." *arXiv preprint arXiv:1806.09055* (2018).

[6] Frankle, Jonathan, and Michael Carbin. "The lottery ticket hypothesis: Finding sparse, trainable neural networks." *arXiv preprint arXiv:1803.03635* (2018).

[7] Wortsman, Mitchell, Ali Farhadi, and Mohammad Rastegari. "Discovering neural wirings." *Advances in Neural Information Processing Systems* 32 (2019).