Trends in One-Shot Neural Architecture Search

Seok Bin Son[†], Soohyun Park[§], and Joongheon Kim[†]
[†]Department of Electrical and Computer Engineering, Korea University, Seoul, Republic of Korea
[§]Division of Computer Science, Sookmyung Women's University, Seoul, Republic of Korea
E-mail: lydiasb@korea.ac.kr, soohyun.park@sookmyung.ac.kr,
joongheon@korea.ac.kr

***Abstract*—Recent advances in neural architecture search (NAS) techniques have provided effective solutions for managing high-dimensional data. These methods algorithmically automate the design process of NAS, aiming to identify an optimized structure that meets target performance criteria with reduced time and resource expenditure. This paper introduces the One-shot NAS method, which offers a computationally efficient approach to exploring NAS while minimizing the associated computational costs.**

I. Introduction

In recent years, deep neural networks (NNs) have achieved remarkable success in handling high-dimensional data and have significantly advanced various computer vision tasks such as image classification, object recognition, and super-resolution image generation [1]. This success is primarily attributed to the architecture of the neural network. The design of the neural network architecture is crucial for effectively extracting relevant features for specific tasks and attaining optimal performance. A standard deep neural network comprises an input layer, several hidden layers, and an output layer. Therefore, it is vital to carefully design the neural network structure to maximize performance for a given application.

Traditionally, designing an optimal neural network architecture has relied on heuristic methods that draw upon the expertise and experience of machine learning practitioners. This approach involves experts manually designing a network structure, training it, and iteratively refining it based on performance evaluations to determine the most effective configuration. However, these methods are often time-consuming and expensive. To address these challenges, Neural Architecture Search (NAS) techniques have been developed to automate the design process. NAS algorithms systematically search for an optimized neural network architecture that achieves desired performance with reduced time and cost, positioning themselves as a fundamental technology in automated machine learning (AutoML) [2].

Early NAS algorithms employed evolutionary algorithms, reinforcement learning, and gradient descent techniques to discover neural network structures that surpass those designed by machine learning experts [3]–[5]. However, these methods involve iterative sampling and training of numerous neural network architectures, which leads to substantial exploration costs, particularly as the search space expands [6]–[8]. Consequently, considerable research has been focused on mitigating these exploration costs. For instance, one-shot NAS techniques have been introduced to either reuse weights learned through reinforcement learning for various candidate architectures or to share weights among multiple neural networks, thereby reducing the computational expense of the search process. The one-shot technique is a training approach that integrates multiple candidate neural networks into a single comprehensive network, known as a supernet. This supernet allows for exploring different sub-network architectures to identify the optimal neural network structure [9]. This method is efficient because it enables the evaluation of multiple neural network configurations within a single training process, rather than requiring separate training for each individual network.

In this paper, we examine one-shot methods. Section 2 provides a detailed definition of NAS, Section 3 introduces one-shot methods, and Section 4 presents the conclusions of our investigation.

II. NAS

Traditionally, the machine learning experts have manually designed neural network architectures and iteratively validated their performance. However, this approach is often inefficient in terms of both time and cost. To address these challenges, the field of NAS has emerged. NAS aims to automatically generate optimized neural network structures for specific tasks, to design efficient deep neural networks while conserving computational resources.

NAS typically involves two main components: the search space and the search strategy. The search space encompasses the range of potential neural network architectures. In contrast, the search strategy involves the methods used to sample, train, and evaluate these candidate networks to identify the optimal architecture. The effectiveness of NAS is largely determined by the definition of the search space and the formulation of the search strategy, making their design critical for discovering the most effective neural network structure.

III. One-shot NAS Method

A. *One-shot NAS*

The one-shot exploration strategy represents a novel approach to enhance the efficiency of NAS by enabling the evaluation of various neural network structures within a single training process. Unlike traditional NAS techniques, which require substantial computational resources to train and assess each neural network architecture individually, the one-shot strategy significantly reduces these costs.

979-8-3503-6463-7/24/$31.00 ©2024 IEEE

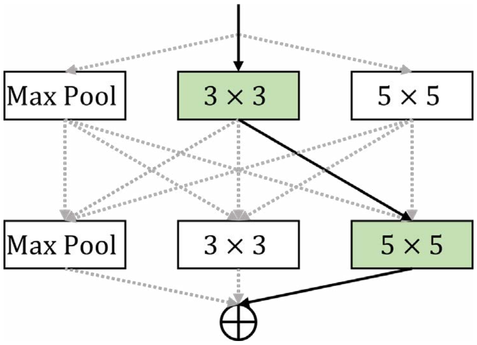

Fig. 1: An example of One-shot NAS.

Central to this strategy is the concept of a supernet. A supernet aggregates all possible neural network structures within a given search space into a single large network, where each path within the network shares the same set of weights. By training the supernet, one can simultaneously evaluate the performance of numerous neural network architectures, thereby minimizing the overall computational effort required.

During the training of the supernet, the entire model is trained, incorporating all possible paths, although only a subset of these paths is activated to optimize computational efficiency. When exploring specific subnets, only certain paths are activated to assess their performance. For instance, as shown in Fig. 1, if a decision must be made between max pooling, 3×3, and 5×5 convolutional operations at a given location, all three operations are included in the supernet. Performance is then evaluated by selectively enabling the specific operation of interest.

This one-shot exploration strategy is increasingly prevalent in contemporary NAS research due to its ability to effectively explore various architectures while making efficient use of computational resources.

B. *ProxylessNAS*

ProxylessNAS is a one-shot NAS methodology designed to efficiently explore and identify high-performance deep learning models [10]. While effective in discovering optimal network structures, traditional NAS techniques are often hindered by significant computational costs. To mitigate this, some NAS methods have utilized smaller proxy models to simplify the exploration process. However, these proxy-based approaches have limitations, particularly in addressing optimization challenges and fully accounting for latency issues.

ProxylessNAS addresses these limitations by directly exploring the network structure while reducing computational overhead. It employs a strategy of pruning unnecessary paths within the one-shot model to minimize both memory usage and computational load. However, when considering all possible paths during training, memory requirements increase proportionally with the number of paths.

To address this issue, ProxylessNAS introduces a binarized gate that activates only one path at a time during runtime, ensuring that only the operations associated with the selected path are executed. Additionally, ProxylessNAS includes a structure parameter representing each candidate operation's selection probability. This parameter is updated alongside the candidate operation parameters, allowing for the pruning of operations with low selection probabilities, thus streamlining the final path discovery.

Moreover, ProxylessNAS incorporates latency as a differentiable component within the loss function. This integration enables direct optimization of latency during the learning process by calculating the expected latency based on the selection probabilities of candidate operations and their associated latencies. This approach allows ProxylessNAS to efficiently explore optimal network structures that balance performance with computational efficiency.

IV. Concluding Remarks

This paper examines the NAS technique, developed to enhance the efficiency of neural network design, traditionally dependent on the expertise of machine learning professionals. NAS automates the process of identifying optimal network architectures, thereby streamlining the design process. Despite its advantages, conventional NAS methods are hindered by substantial computational costs and time demands. To overcome these challenges, this paper explores the One-shot NAS approach. One-shot NAS allows for the simultaneous evaluation of multiple network architectures within a single training process, significantly reducing computational expenses while enhancing the efficiency of the search process.

V. Acknowledgement

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2021-0-00766, Development of Integrated Development Framework that supports Automatic Neural Network Generation and Deployment optimized for Runtime Environment).

References

[1] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, no. 1, p. 7068349, 2018.

[2] Y. Kim, W. J. Yun, Y. K. Lee, S. Jung, and J. Kim, "Trends in neural architecture search: Towards the acceleration of search," in *Proc. of the International Conference on Information and Communication Technology Convergence, ICTC*, Jeju Island, Korea, Republic of, October 2021, pp. 421–424.

[3] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. of the International Conference on Learning Representations, ICLR*, Toulon, France, April 2017.

[4] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Salt Lake City, UT, USA: Computer Vision Foundation / IEEE Computer Society, June 2018, pp. 8697–8710.

[5] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proc. of the AAAI Conference on Artificial Intelligence, AAAI*, Honolulu, Hawaii, USA, January 2019, pp. 4780–4789.

[6] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, "Efficient neural architecture search via parameter sharing," in *Proc. of the International Conference on Machine Learning, ICML*, ser. Proceedings of Machine Learning Research, vol. 80. Stockholmsmässan, Stockholm, Sweden: PMLR, July 2018, pp. 4092–4101.

[7] H. Liu, K. Simonyan, and Y. Yang, “DARTS: differentiable architecture search,” in *Proc. of the International Conference on Learning Representations, ICLR*. New Orleans, LA, USA: OpenReview.net, May 2019.

[8] Y. Kim, S. Jung, M. Choi, and J. Kim, “Search space adaptation for differentiable neural architecture search in image classification,” in *Proc. of the International Conference on Ubiquitous and Future Networks, ICUFN*. Barcelona, Spain: IEEE, July 2022, pp. 363–365.

[9] G. Bender, P. Kindermans, B. Zoph, V. Vasudevan, and Q. V. Le, “Understanding and simplifying one-shot architecture search,” in *Proc. of the International Conference on Machine Learning, ICML 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80, Stockholmsmässan, Stockholm, Sweden, July 2018, pp. 549–558.

[10] H. Cai, L. Zhu, and S. Han, “Proxylessnas: Direct neural architecture search on target task and hardware,” in *Proc. of the International Conference on Learning Representations, ICLR*. New Orleans, LA, USA: OpenReview.net, May 2019.