ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10094709

RD-NAS: ENHANCING ONE-SHOT SUPERNET RANKING ABILITY VIA RANKING DISTILLATION FROM ZERO-COST PROXIES

Peijie Dong[1], *Xin Niu*[1,*], *Lujun Li*[2], *Zhiliang Tian*[1,*], *Xiaodong Wang*[1]
Zimian Wei[1], *Hengyue Pan*[1], *Dongsheng Li*[1]

[1] School of Computer, National University of Defense Technology, Hunan, China
[2] Chinese Academy of Sciences, Beijing, China

ABSTRACT

Neural architecture search (NAS) has made tremendous progress in the automatic design of effective neural network structures but suffers from a heavy computational burden. One-shot NAS significantly alleviates the burden through weight sharing and improves computational efficiency. Zero-shot NAS further reduces the cost by predicting the performance of the network from its initial state, which conducts no training. Both methods aim to distinguish between "good" and "bad" architectures, i.e., ranking consistency of predicted and true performance. In this paper, we propose Ranking Distillation one-shot NAS (RD-NAS) to enhance ranking consistency, which utilizes zero-cost proxies as the cheap teacher and adopts the margin ranking loss to distill the ranking knowledge. Specifically, we propose a margin subnet sampler to distill the ranking knowledge from zero-shot NAS to one-shot NAS by introducing Group distance as margin. Our evaluation of the NAS-Bench-201 and ResNet-based search space demonstrates that RD-NAS achieve 10.7% and 9.65% improvements in ranking ability, respectively. Our codes are available at https://github.com/pprp/CVPR2022-NAS-competition-Track1-3th-solution

Index Terms— neural architecture search, one-shot NAS, zero-shot NAS, rank consistency

1. INTRODUCTION

Neural Architecture Search (NAS) has sparked increased interest due to its remarkable progress in a variety of computer vision tasks [5, 3]. It aims to reduce the cost of human efforts in manually designing network architectures and discover promising models automatically. Early NAS works [6, 7] take thousands of GPU hours in search cost, and ENAS [8] first attempt at weight-sharing techniques to accelerate the searching process. The key to weight-sharing-based methods is an over-parameterized network, *a.k.a.* supernet, that encompasses all candidate architectures in the search space. The weight-sharing-based NAS approaches [1, 9] are called one-shot NAS

*Corresponding author

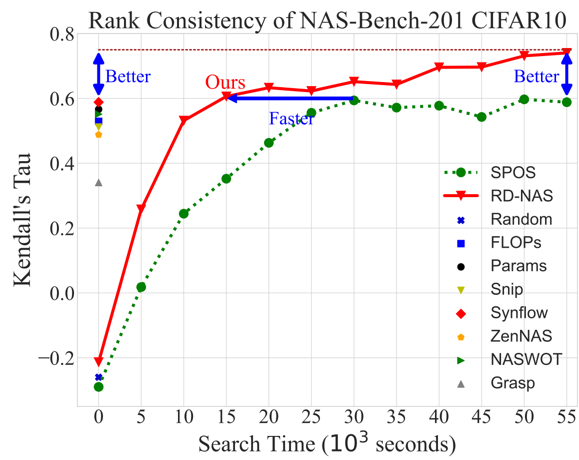

Fig. 1: The ranking consistency of one-shot NAS (SPOS [1]) and zero-shot NAS(FLOPs [2], Params [2], Snip [2], Synflow [2], ZenNAS [3], NASWOT [4], Grasp [2]). Our proposed RD-NAS achieves higher Kendall's Tau than both one-shot and zero-shot NAS and converges faster than the baseline [1].

since it only requires the cost of training one supernet. Despite the high efficiency of one-shot NAS, it is theoretically based on the ranking consistency assumption, *a.k.a.* the estimated performance of candidate architectures in supernet should be highly correlated with the true performance of the corresponding architecture when trained from scratch. However, due to the nature of the weight-sharing approach, the subnet architectures interfere with each other, and the estimated accuracy from the supernet is inevitably averaged.

A recent trend of NAS focuses on zero-cost proxies [4, 2], which aims to estimate the relative performance of candidate architecture from a few mini-batch of data without training the model. The approach of inferring the trained accuracy directly from the initial state of the model is called zero-shot NAS since it does not involve any training. However, there is a clear preference for zero-shot NAS that affects its ability to find good architectures [10].

To alleviate the ranking disorder caused by weight-sharing, some predictor-based methods [11] use the ranking loss to enhance the ranking ability of the model, but the challenge is that acquiring training samples (pairs of (model, accuracy)) is computationally expensive. To improve the ranking consistency