Improve Ranking Correlation of Super-net through Training Scheme from One-shot NAS to Few-shot NAS

Jiawei Liu,* Kaiyu Zhang,* Weitai Hu,* and Qing Yang
Du Xiaoman Financial, Beijing, China
{liujiawei, zhangkaiyu, huweitai, yangqing}@duxiaoman.com

Abstract

The algorithms of one-shot neural architecture search (NAS) have been widely used to reduce the computation. However, because of the interference among the subnets which weights are shared, the subnets inherited from these super-net trained by those algorithms have poor consistency in precision ranking. To address this problem, we propose a step-by-step training super-net scheme from one-shot NAS to few-shot NAS. In the training scheme, we training super-net by the one-shot way firstly, and then we disentangles the weights of super-net by splitting that to multi-subnets and training them gradually. Finally, our method ranks 4th place in the CVPR2022 Lightweight NAS Challenge Track1. Our code is available at https://github.com/liujiawei2333/CVPR2022-NAS-competition-Track-1-4th-solution.

1. Introduction

Neural architecture search (NAS) aims to automatically design neural network architectures. As one of the widely-used NAS methods, one-shot NAS utilizes a super-net subsuming all candidate architectures (subnets) to implement NAS function. All subnets directly inherit their weights from the super-net which is only trained once. One-shot NAS provides a large number of subnets for a single training session at a low training cost.

NAS methods such as SPOS [4] and FairNAS [2] independently connect all candidates of the operation to construct a super-net, sampling one or more subnets and training them in each training iteration. On this basis, OFA [1] constructs the super-net through entangling the weights of candidates with inclusion relations together, and the super-net is trained by progressive shrinking method and knowledge distillation. On the basis of the super-net constructed by OFA, BigNAS [9] trains the super-net at one time through sandwich rule and several tricks and no longer

needs to fine-tune the subnet. However, these method will cause interference among the subnets, so that the accuracy ranking of subnets is inconsistence with the one of subnets when they are trained from scratch independently.

Some methods attempt to improve the ranking correlation. Pi-NAS [6] reduces the consistency problem to feature offset and parameter offset, then solves the problem through self-supervised learning scheme of cross-path training super-net. Landmark Regularization [10] takes the accuracy of a few standalone trained subnets as regularization to guide the training of super-nets. CVPR2021 3rd Lightweight NAS Challenge track1 [3] shows the influence on different aspects, including super-net construction, super-net training and sampling on ranking correlation, which draws empirical conclusions through a large number of experiments. CVPR2021 1st Lightweight NAS Challenge track1 [8] maps the super-net to the high-capacity super-net in stages and adds narrow branches of the super-net. Few-shot NAS [13] randomly selects different candidate as one group to split the search space, thus generating multiple sub-super-nets. GM-NAS [5] uses the gradient matching method to propose a few-shot NAS judgment benchmark for space segmentation. K-Shot NAS [7] trains k super-nets and samples subnets by combining the weights of these super-nets.

There are also two studies that bring new thinking. [11] designs a small search space and finds a phenomenon: the ranking correlation varies greatly in different epochs and is even unstable in different iterations of the same epoch. [12] also finds several interesting conclusions: the increase of epoch could hardly improve the ranking correlation, and the accuracy of top-K subnets has little relationship with ranking correlation. In addition, the ranking correlation is strongly correlated with the search space.

In this paper, we propose a step-by-step training scheme for super-net from one-shot NAS to few-shot NAS. We believe that reducing the degree of weight entanglement among subnets is the key to improve the consistency of super-net, so we gradually split the weight of one-shot super-net so that make it decoupled to few-shot super-net,

---

*Equal Contribution. Listing order is random.

and then we follow knowledge distillation and sandwich rule to gradually train the super-net. From one-shot super-net to few-shot super-net, the weight of super-net is inherited step by step. In addition, we use adaptive learning rate to alleviate the problem of different frequency of module sampling in few-shot super-net.

2. Proposed method

2.1. Search Space

Different from CVPR2021 Lightweight NAS Challenge Track1, which only searches for the number of channels, this track in 2022 searches for the number of layers of the network and the number of channels at each layer. It would be very difficult to construct a super-net like what SPOS [4] and FairNAS [2] have done, so we construct that according to the ways of OFA [1] and BigNAS [9] with weight entanglement. The super-net is constructed based on resnet48, in which contains 4 stages. The number of searchable channels is the product of basic channel and scaling ratio. The scaling ratio in all stages is [1.0, 0.95, 0.9, 0.85, 0.8, 0.75, 0.7]. The number of searchable blocks is from 2 to 8. The details of search space in every stage is shown in Table 1.

Stage	Block	Basic channels
1	[2,3,4,5]	64
2	[2,3,4,5]	128
3	[2,3,4,5,6,7,8]	256
4	[2,3,4,5]	512

Table 1. The search space of CVPR 2022 Lightweight NAS challenge track1

2.2. Training super-net

For the one-shot super-net, we follow the pipeline of BigNAS [9]. In details, we first construct the super-net based on the search space of S with weights $\boldsymbol{\theta}$, and then we train the largest sub-network S_{max} with the cross entropy loss $\mathcal{L}_{CE}$. Finally, we distill the smallest sub-network S_{min} and two sub-networks sampled randomly respectively with the Kullback–Leibler (KL) loss (distance) $\mathcal{L}_{KL}$ between the logit of them and the one of S_{max}. Algorithm 1 shows the training process.

As for few-shot super-net, in order to alleviate the interference among the subnets whose weights are sharing, the proposed method follows the way of few-shot NAS, which disentangles the shared weights by splitting the super-net to multi-subnet during the training process. In this section, we introduce the split way for the resnet-like search space.

There are two kinds of channel number of convolution layers need to be searched in one block, in which the second convolution layer controls the channel number of the output features. Because of the existence of identity mapping in the block of the resnet, the channel number of the output features is the same in every block in the same stage. In other words, the second channel number in different blocks in the same stage is fixed in once search process. Based on this analysis, we divide the search space following the dimensionality of the number in second convolution layer. Specifically, the candidates of the second channel number is 88, 96, 104, 112, 120 and 128 for the second stage of this search space, whose basic channel number is 128. Instead of retain only one group based on the vanilla one-shot NAS, we extend the number of the group up to the number of the candidates, that is to say, we divide the second stage into six groups, each group contains the same architecture except for the channel number of the second convolution layer. We apply this split way to all stages except for the stem stage. The left of Figure 1 shows the architecture details of all stages.

According to the aforementioned ways, we first train the super-net with one group by Algorithm 1 until convergence (the same as the one-shot way), and then the super-net with two groups is trained following the same algorithm with the weights inheriting from the ones of the super-net with one group (The details of weights inheriting can be seen in the right of Figure 1). We repeat the training process until the number of the group increasing to maximum.

Algorithm 1 One-shot Super-net Training

Input: super-net S, weights $\boldsymbol{\theta}$, total training step T, loss $\mathcal{L}_{CE}$ and $\mathcal{L}_{KL}$.

Output: the optimal weights $\boldsymbol{\theta}^*$.

1: fix random seed.
2: random initialize $\boldsymbol{\theta}$.
3: **for** $t \leftarrow 1, T$ **do**
4:     set all gradients to zeros for super-net S.
5:     train the largest sub-net S_{max} with loss $\mathcal{L}_{CE}$.
6:     calculate gradients for super-net S based on $\mathcal{L}_{CE}$.
7:     **for** $k \leftarrow 1, 3$ **do**
8:         **if** $k \neq 3$ **then**
9:             random sample sub-net S_{sample};
10:         **else**
11:             sample the smallest sub-net as S_{sample};
12:         **end if**
13:         distill the S_{sample} with the outputs of S_{max};
14:         calculate gradients based on $\mathcal{L}_{KL}$.
15:     **end for**
16:     update $\boldsymbol{\theta}$ by accumulated gradients.
17: **end for**
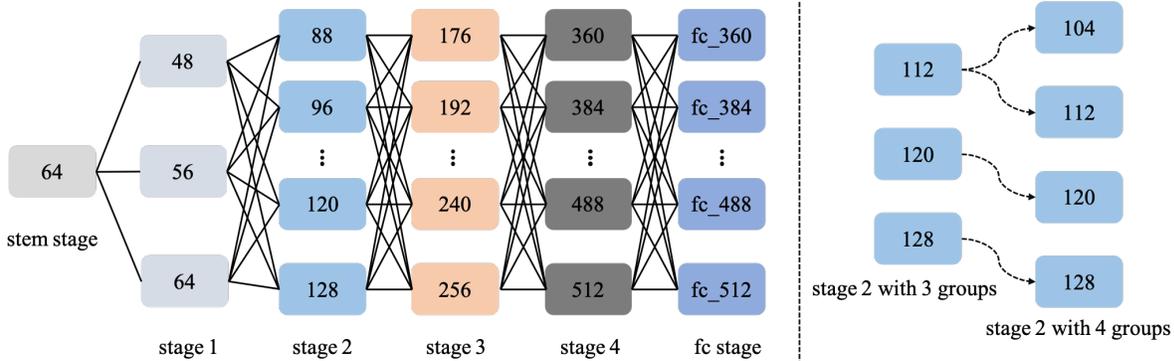18: return the optimal weights $\boldsymbol{\theta}^*$.

Figure 1. **Left:** The super-net architecture with the largest group numbers. The box with number represents one group with the number as the output-feature channel. The solid-black line stands for the flow of features. From stem stage to full-connection stage, each stage includes 1, 3, 6, 7, 7, 7 groups respectively. **Right:** The inheriting method. Taking stage 2 as an example, the dashed-black arrow stands for the source of the weights for the 4 groups in second stage. Especially, the weights of second convolution layer in the group with 104 output channels is replica from the weights of the same layer containing the former 104 channels in the old group with 112 output channels.

3. Experiments

3.1. Results of one/few-shot super-net

We firstly train the one-shot super-net for 100 epoches with batch size 128 and apply a stochastic gradient descent optimizer with the momentum of 0.9. The learning rate (LR) is decreased from an initial value of 0.1 to 0 with a linear-warmup cosine learning rate decay strategy, and warmup epoch is 5. For the largest subnet, the weights are regularized with weight decay of 0.00001, and dropout probability is 0.2. The data augmentation includes brightness and contrast changes with a probability of 0.5, random rotation within 15 degrees, and random clipping of 224*224.

The training configurations of few-shot NAS are mostly the same as that of one-shot NAS, except that epoch is set to 20 and there is no warm-up for learning rate. In addition, different network modules are assigned different learning rates according to their probability of being sampled, to ensure the same order of magnitude in the training times. As shown in Table 2, taking group 3 as an example, the LR of module with channel number 112 in stage 1 is 4 times of basic LR. LR of module with channel number 232 in stage 2 and channel number 464 in stage 3 is 5 times of basic LR, and the learning rate of the rest modules are basic learning rate.

We selected the corresponding model at each training super-net stage to calculate the score, as shown in Table 3.

3.2. Ranking changes while training

As [11] said, ranking does not increase steadily during training as accuracy does. Therefore, it is important to select the most appropriate epoch for the super-net. However, the large test set and the large number of subnets to be eval-

Group	Stage	Channels	LR
	1	64	2x
2	2	128	5x
	3,4	256,512	6x
3	2	112	4x
	3,4	232,464	5x
4	2	104	3x
	3,4	216,432	4x
5	2	96	3x
	3,4	208,408	4x
6	2	88	3x
	3,4	192,384	4x

Table 2. Under different group of few-shot NAS, the LR corresponding to different modules

Stage	Group	Epoch	Pearson Coeff.
One-shot		80	0.80821
	2	11	0.81008
	3	17	0.81759
	4	19	0.81547
Few-shot	5	18	0.81463
	6	4	**0.81830**
	7	11	0.81521

Table 3. The ranking correlation of subnets inherit their weights from super-net

uated make it time-consuming to calculate rankings on each epoch. To solve this problem, we standalone trained 30 subnets with the part of valid datasets (the amount is $\frac{1}{20}$ of the original datasets') to obtain the target accuracy, and then

used the same datasets to evaluate the accuracy of the 30 subnets inherited from the weight of the current super-net after each epoch when training the super-net. Therefore, we obtained the change curve of Kendall tau with the epoch as shown in Figure 2, which shows that the ranking is rising but very unstable. The super-net with late epoch and large Kendall tau is selected as the measured model.

Figure 2. Kendall tau curve changed with epoch.

3.3. Ablation on Architecture Sampling Strategies

We discuss the different ranking performance of super-nets based on different architecture sampling strategies in this paragraph. To reduce the test time, we evaluate the super-net trained with one-shot method on the part of datasets mentioned before. Table 4 shows the results about the uniform sample and fair sample, in which the former is better than latter obviously, opposite to the results of [2]. This may be caused by the wrong usage of the fair samples. In [2], it requires the same amount of candidates in search space, which is not satisfied strictly in this search space.

Sample	Pearson Coeff.
Uniform	0.6093
Fair	0.42489

Table 4. The results of different sampling strategies

4. Conclution

In this paper, with the training scheme in combination with one-shot NAS and few-shot NAS, we improve the ranking consistency of the subnets whose weights are inherited from the ones of super-net. In detail, we first train the super-net with the one-shot method, and then to disentangle the weights, we extend the stages of super-net from one group up to the maximum along with the weights inhering. Finally, the different stages in super-net with different groups use the adaptive LR for sufficient training. In addition, we investigate the relationship between the consistency of subnets and the epochs, as well as the influences of the sample strategies of subnets on the ranking correlation.

References

[1] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once for all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020.

[2] Xiangxiang Chu, Bo Zhang, and Ruijun. Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search. In *International Conference on Computer Vision*, 2021.

[3] Chaoyu Guan, Yijian Qin, Zhikun Wei, Zeyang Zhang, Zizhao Zhang, Xin Wang, and Wenwu Zhu. One-shot neural channel search: What works and what's next.

[4] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. In *European Conference on Computer Vision*, pages 544–560. Springer, 2020.

[5] Shoukang Hu, Ruochen Wang, HONG Lanqing, Zhenguo Li, Cho-Jui Hsieh, and Jiashi Feng. Generalizing few-shot nas with gradient matching. In *International Conference on Learning Representations*, 2022.

[6] Jiefeng Peng, Jiqi Zhang, Changlin Li, Guangrun Wang, Xiaodan Liang, and Liang Lin. Pi-nas: Improving neural architecture search by reducing supernet training consistency shift. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12354–12364, 2021.

[7] Xiu Su, Shan You, Mingkai Zheng, Fei Wang, Chen Qian, Changshui Zhang, and Chang Xu. K-shot nas: Learnable weight-sharing for nas with k-shot supernets. In *International Conference on Machine Learning*, pages 9880–9890. PMLR, 2021.

[8] Ziwei Yang, Ruyi Zhang, Zhi Yang, Xubo Yang, Lei Wang, and Zheyang Li. Improving ranking correlation of supernet with candidates enhancement and progressive training. *arXiv preprint arXiv:2108.05866*, 2021.

[9] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Bender Gabriel, Pieter-Jan Kindermans, Mingxing Tan, Thomas Huang, Xiaodan Song, Ruoming Pang, and Quoc Le. Bignas: Scaling up neural architecture search with big single-stage models. In *European Conference on Computer Vision*, pages 702–717. Springer, 2020.

[10] Kaicheng Yu, Rene Ranftl, and Mathieu Salzmann. Landmark regularization: Ranking guided super-net training in neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13723–13732, 2021.

[11] Yuge Zhang, Zejun Lin, Junyang Jiang, Quanlu Zhang, Yujing Wang, Hui Xue, Chen Zhang, and Yaming Yang. Deeper insights into weight sharing in neural architecture search. *arXiv preprint arXiv:2001.01431*, 2020.

[12] Yuge Zhang, Quanlu Zhang, and Yaming Yang. How does supernet help in neural architecture search? *arXiv preprint arXiv:2010.08219*, 2020.

[13] Yiyang Zhao, Linnan Wang, Yuandong Tian, Rodrigo Fonseca, and Tian Guo. Few-shot neural architecture search. In *International Conference on Machine Learning*, pages 12707–12718. PMLR, 2021.