Breaking the Curse of Space Explosion: Towards Efficient NAS with Curriculum Search

Yong Guo[1], Yaofo Chen[1], Yin Zheng[2], Peilin Zhao[3],
Jian Chen[1 4], Junzhou Huang[5] , Mingkui Tan[1 6]

[1]School of Software Engineering, South China University of Technology
[2]Weixin Group, Tencent [3]Tencent AI Lab , Tencent
[4]Guangdong Key Laboratory of Big Data Analysis and Processing
[5]University of Texas at Arlington [6]Pazhou Laboratory

Contents

Contents

Background

Deep neural networks have been producing state-of-the-art results in many challenging tasks, such as image classification, object detection, semantic segmentation and etc.

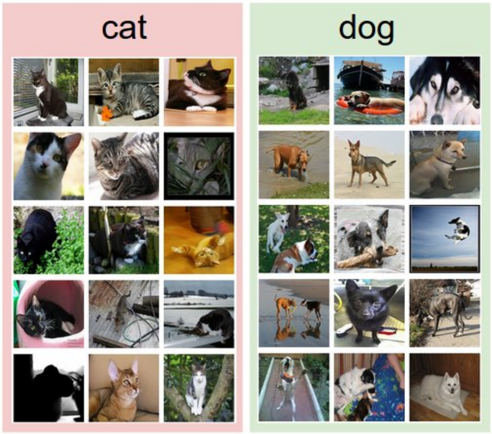

Image Classification

Object Detection

Semantic Segmentation

Figure: Applications of deep neural networks.

Neural Architecture Design

- Neural architecture design is one of the key factors behind the success of deep neural networks.

- Existing architectures can be divided into two categories:

  1. Manually designed architectures
  2. Automatically searched architectures by Neural Architecture Search (NAS)

- Empirical studies show that the automatically searched architectures often outperform the manually designed ones.

Search Space Size Analysis

Space Explosion Issue

The search space in NAS is often **extremely large**.

Given B nodes and K candidate operations in a cell-based architecture, the size of the search space Ω can be computed by

$$|\Omega| = K^{2(B-3)}\left((B-2)!\right)^2$$

- ENAS has a search space size of 5×10^{12} with B=8 and K=5
- DARTS has a search space size of 2×10^{11} with B=7 and K=8.

Search Space Size Analysis

- As the number of nodes/operations increases, the size of the search space will increase.
- Increasing nodes make the size of search space grow faster than increasing operations.

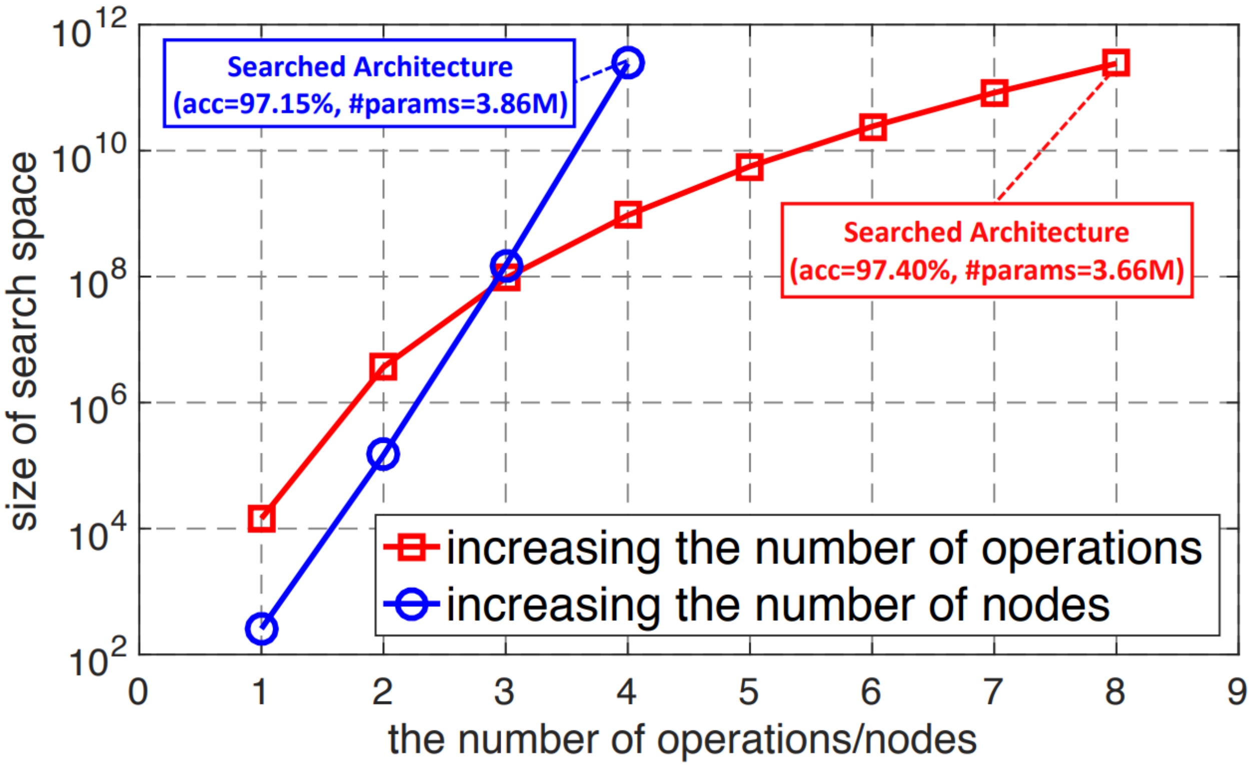

Figure: Comparisons of the search spaces size of different number of operations/nodes.

Motivation

To alleviate the space explosion issue, we seek to enlarge the search space gradually to improve the search performance by curriculum learning.

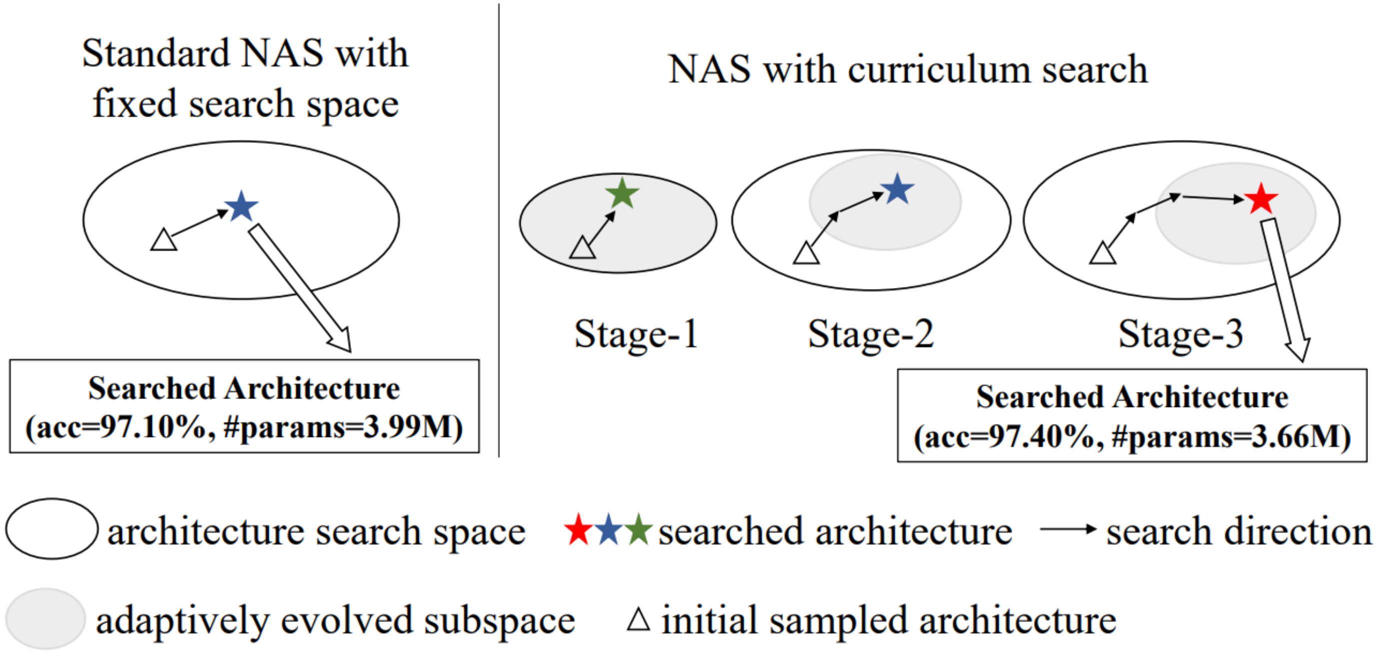

Figure: Comparisons of the search process between standard NAS methods and our proposed curriculum NAS method..

Contents

Preliminary

Reinforcement Learning (RL) based NAS methods seek to learn a controller to produce candidate architectures.

$$\max_{\theta} \mathbb{E}_{\alpha\sim\pi(\alpha;\theta,\Omega)}\mathcal{R}\left(\alpha, w^*(\alpha)\right)$$
$$\text{s.t. } w^*(\alpha) = \arg\min_{w} \mathcal{L}\left(\alpha, w\right)$$

- θ is the parameter of the controller.
- Ω is the search space.
- $R\left(\alpha, w^*(\alpha)\right)$ is some metric to measure the performance of architecture α.
- $\mathcal{L}$ is the loss function on training data.

Curriculum Neural Architecture Search

We propose a novel Curriculum Neural Architecture Search (CNAS) to enlarge the search space by gradually **increasing the number of candidate operations** from 1 to K .

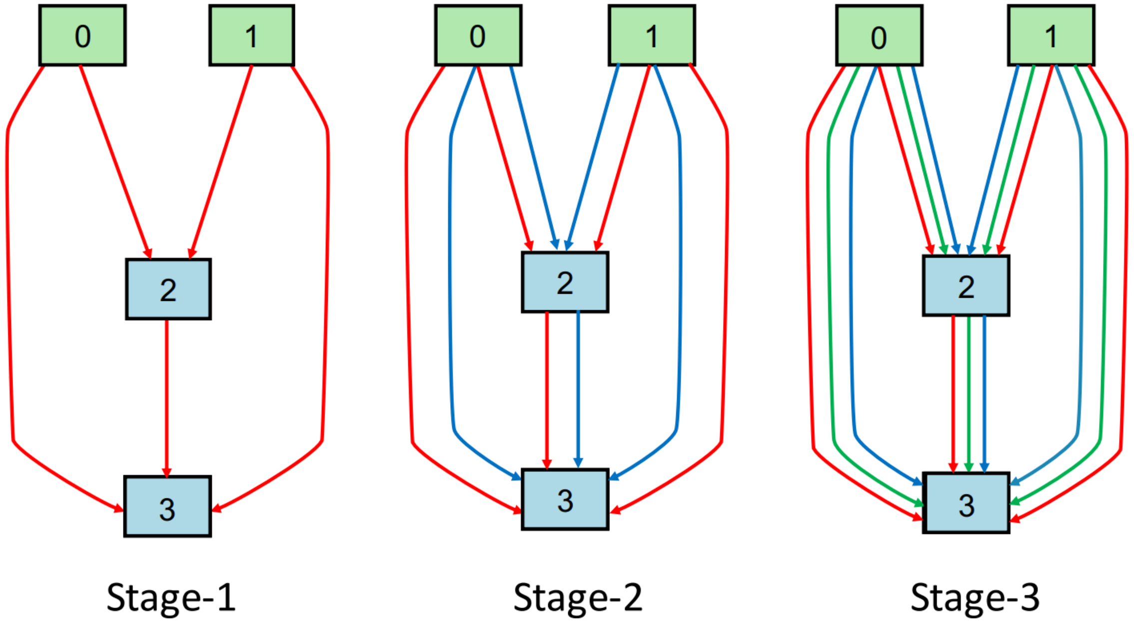

Figure: An overview of the search space used by CNAS.

NAS with Curriculum Search

- The training process can be divided into K stages corresponding to K candidate operations.
- The training objective in *i-th* stage can be written as

$$\max_{\theta} \ \mathbb{E}_{\alpha\sim\pi(\cdot;\theta,\Omega_i)}\left[\mathcal{R}\left(\alpha, w^*(\alpha)\right)\right] + \lambda H\left(\pi\left(\cdot;\theta,\Omega_i\right)\right)$$

$$\text{s.t. } w^*(\alpha) = \arg\min_{w} \mathcal{L}\left(\alpha, w\right),$$

- Ω_i is the search space of the *i-th* stage.
- $\pi(\cdot;\theta,\Omega_i)$ denotes the learned policy w.r.t. Ω_i .
- $H(\cdot)$ evaluates the entropy of the policy.
- λ controls the strength of the entropy regularization term.

Operation Warmup

Operation Unfairness

The architectures with the new operation have very poor performance.

We propose an **operation warmup** method.

- We fix the controller model and only train the parameters of the super network.
- We uniformly sample candidate architectures to train each operation with equal probability.

The architectures with the newly added operation achieve **comparable performance** with the architectures without this operation.

Training Method

Algorithm 1 Training method for CNAS.

Require: The operation sequence O, learning rate η, the number of the iterations for operation warmup M, the uniform distribution of architectures $p(\cdot)$, the controller's policy $\pi(\cdot)$, super network parameters w, controller parameters θ.

Initialize w and θ, $\Omega_0 = \emptyset$.
for i=1 to $|O|$ **do**
    Enlarge Ω_i by adding O_i to the set of candidate operations;
    // Operation warmup
    for j=1 to M **do**
        Sample $\alpha \sim p(\alpha; \Omega_i)$;
        $w \leftarrow w - \eta \nabla_w \mathcal{L}(\alpha, w)$;
    end for
    while not convergent **do**
        // Update θ by maximizing the reward
        for each iteration on validation data **do**
            Sample $\alpha \sim \pi(\alpha; \theta, \Omega_i)$;
            Update the controller by ascending its gradient:
                $\mathcal{R}(\alpha, w) \nabla_\theta \log \pi(\alpha; \theta, \Omega_i) + \lambda H(\pi(\cdot; \theta, \Omega_i))$;
        end for
        // Update w by minimizing the training loss
        for each iteration on training data **do**
            Sample $\alpha \sim \pi(\alpha; \theta, \Omega_i)$;
            $w \leftarrow w - \eta \nabla_w \mathcal{L}(\alpha, w)$.
        end for
    end while
end for

Contents

Demonstration of CNAS

- **Fixed-NAS**: For each stage, we keep the search space fixed and train a controller from scratch.
- **CNAS**: We train the controller in a growing search space by gradually adding new operations.
- **CNAS-Node**: We train the controller in a growing search space by gradually adding new nodes.

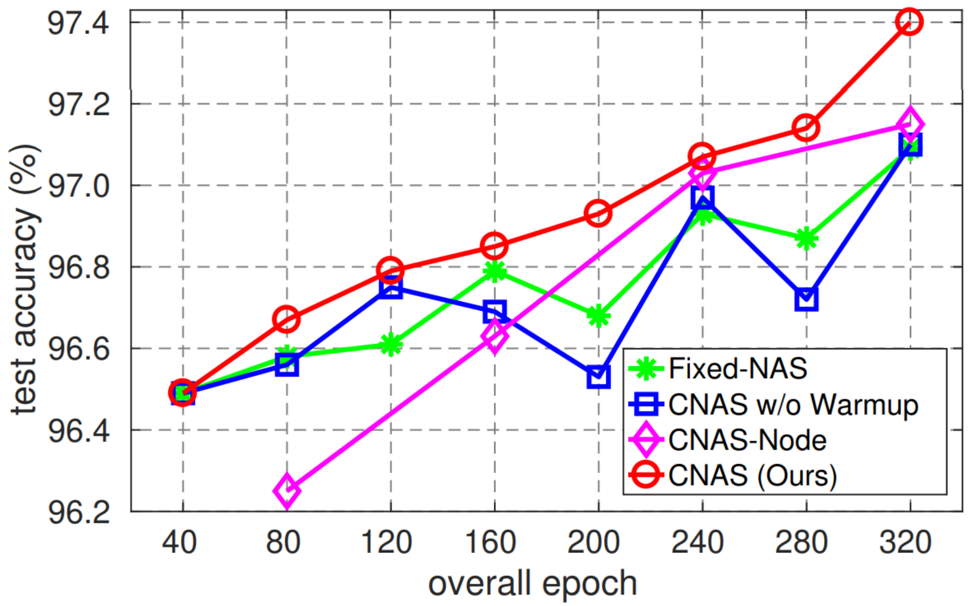

Figure: Performance comparisons of the architectures obtained by different methods during the search process.

Evaluation on CIFAR-10

CNAS yields significantly better performance than the baseline architectures on CIFAR-10.

Architecture	Test Accuracy (%)	Params (M)	Search Costs (GPU days)
DenseNet-BC (Huang et al., 2017)	96.54	25.6	–
PyramidNet-BC (Han et al., 2017)	96.69	26.0	–
Random search baseline	96.71 ± 0.15	3.2	–
NASNet-A + cutout (Zoph et al., 2018)	97.35	3.3	1800
NASNet-B (Zoph et al., 2018)	96.27	2.6	1800
NASNet-C (Zoph et al., 2018)	96.41	3.1	1800
AmoebaNet-A + cutout (Real et al., 2019)	96.66 ± 0.06	3.2	3150
AmoebaNet-B + cutout (Real et al., 2019)	96.63 ± 0.04	2.8	3150
DSO-NAS (Zhang et al., 2018b)	97.05	3.0	1
Hierarchical Evo (Liu et al., 2018b)	96.25 ± 0.12	15.7	300
SNAS (Xie et al., 2019)	97.02	2.9	1.5
ENAS + cutout (Pham et al., 2018)	97.11	4.6	0.5
NAONet (Luo et al., 2018)	97.02	28.6	200
NAONet-WS (Luo et al., 2018)	96.47	2.5	0.3
GHN (Zhang et al., 2018a)	97.16 ± 0.07	5.7	0.8
PNAS + cutout (Liu et al., 2018a)	97.17 ± 0.07	3.2	225
DARTS + cutout (Liu et al., 2019)	97.24 ± 0.09	3.4	4
CARS + cutout (Yang et al., 2019)	97.38	3.6	0.4
CNAS + cutout	$\mathbf{97.40 \pm 0.06}$	3.7	0.3

Evaluation on CIFAR-10

CNAS finds better architectures than existing methods on ImageNet.

Architecture	Test Accuracy (%)		#Params (M)	#MAdds (M)	Search Cost (GPU days)
	Top-1	Top-5			
ResNet-18 (He et al., 2016)	69.8	89.1	11.7	1814	–
Inception-v1 (Szegedy et al., 2015)	69.8	89.9	6.6	1448	–
MobileNet (Howard et al., 2017)	70.6	89.5	4.2	569	–
NASNet-A (Zoph et al., 2018)	74.0	91.6	5.3	564	1800
NASNet-B (Zoph et al., 2018)	72.8	91.3	5.3	488	1800
NASNet-C (Zoph et al., 2018)	72.5	91.0	4.9	558	1800
AmoebaNet-A (Real et al., 2019)	74.5	92.0	5.1	555	3150
AmoebaNet-B (Real et al., 2019)	74.0	92.4	5.3	555	3150
GHN (Zhang et al., 2018a)	73.0	91.3	6.1	569	0.8
SNAS (Xie et al., 2019)	72.7	90.8	4.3	522	1.5
DARTS (Liu et al., 2019)	73.1	91.0	4.9	595	4
NAT-DARTS (Guo et al., 2019)	73.7	91.4	4.0	441	-
PNAS (Liu et al., 2018a)	73.5	91.4	5.1	588	255
MnasNet-92 (Tan et al., 2019)	74.8	92.0	4.4	-	-
ProxylessNAS (Cai et al., 2019)	75.1	92.5	7.1	-	8.3
CARS (Yang et al., 2019)	75.2	92.5	5.1	591	0.4
CNAS	**75.4**	**92.6**	5.3	576	**0.3**

Contents

Conclusion

- We propose a novel Curriculum Neural Architecture Search (CNAS) method to alleviate the training difficulties of the NAS problem incurred by the extremely large search space.
- We propose a curriculum search method that gradually incorporates the knowledge learned from a small search space.
- Extensive experiments show the superiority of CNAS over the hand-crafted and NAS based architectures.

Thanks!

Q & A