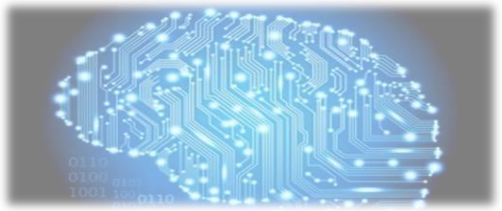
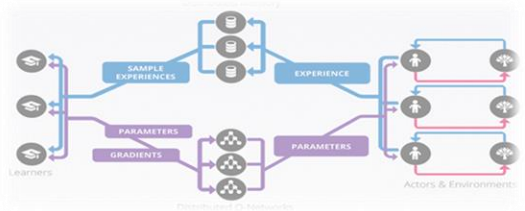



You Only Search Once: Single Shot Neural Architecture Search via Direct Sparse Optimization

AutoML 연구실
석사과정 문아성

21.04.06

You Only Search Once: Single Shot Neural Architecture Search via Direct Sparse Optimization

- Evolution-Based NAS
- Reinforcement Learning-Based NAS
- Gradient-Based NAS

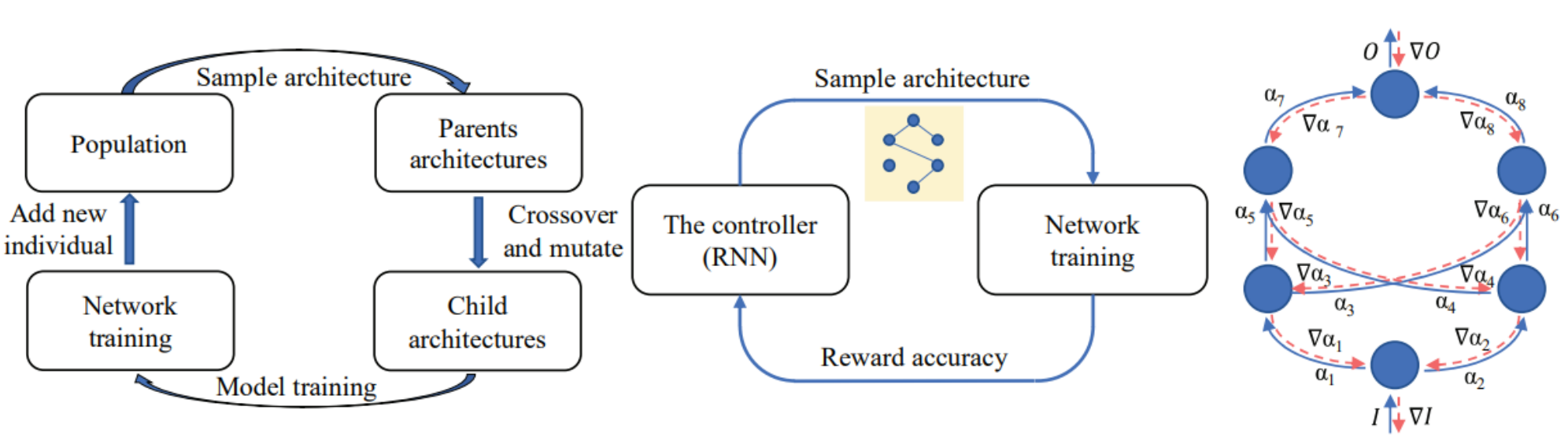


Fig. 1. Three mainstream methods of neural architecture search: (a) Evolution, where a population of architectures is updated based on their fitness, e.g., performance on the validation set. New individuals are produced by crossover and mutate operation.Reinforcement learning, where an RNN controller is applied to sample architectures and trained based on the performance of the architectures. (b) Reinforcement learning, where an RNN controller is applied to sample architectures and optimized based on the performance of the sampled architectures. (c) Gradient based method, where all paths in hypenetwork are parameterized with architecture parameters, both weights and architecture parameters are updated with gradients.

DSO-NAS Architecture

$$\mathbf{h}_{(b,i,j)} = \mathcal{O}_{(b,i,j)} \left(\sum_{m=1}^{i-1} \sum_{n=1}^{N} \mathbf{h}_{(b,m,n)} + \mathbf{O}_{(b-1)} \right)$$

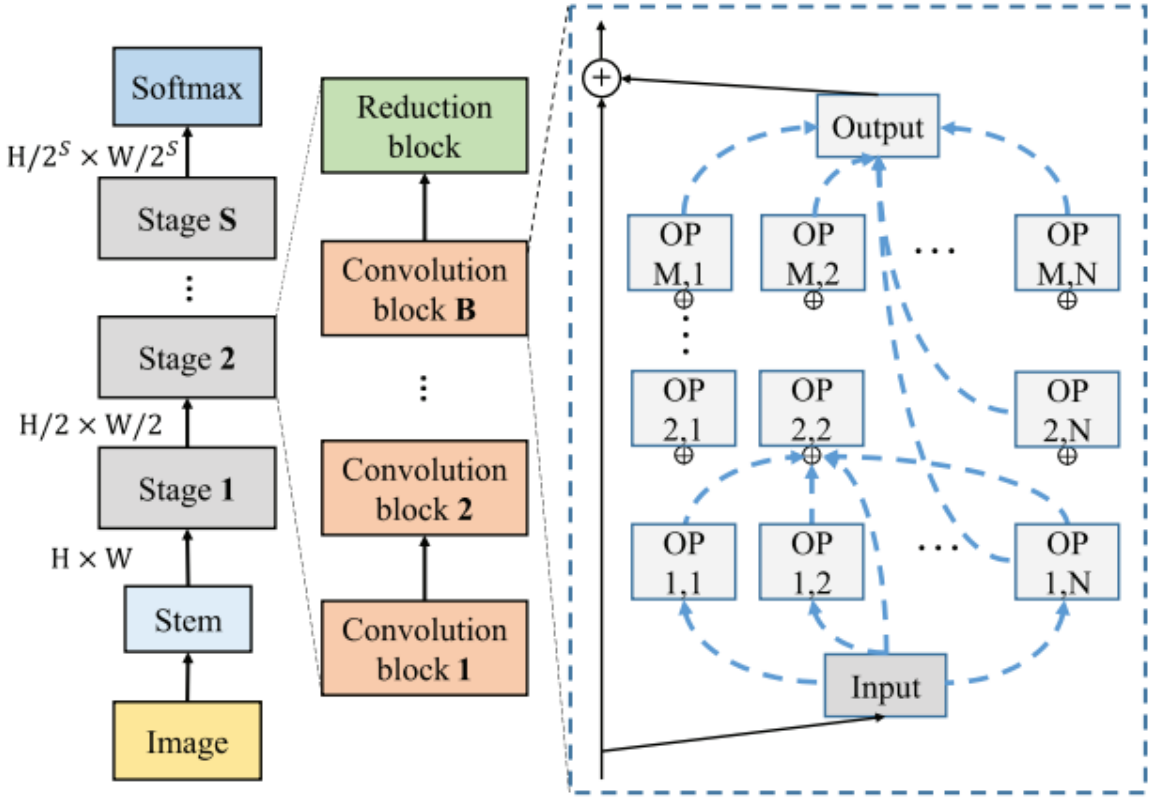


Fig. 2. We divide a network into **S** stages according to the size of feature maps. A stage consists of **B** convolution blocks and ends up with a reduction block. A block composed of M levels and every level contains N different operations. Following [14], we adopt differnet stem structure for CIFAR and ImageNet.

Network Pruning

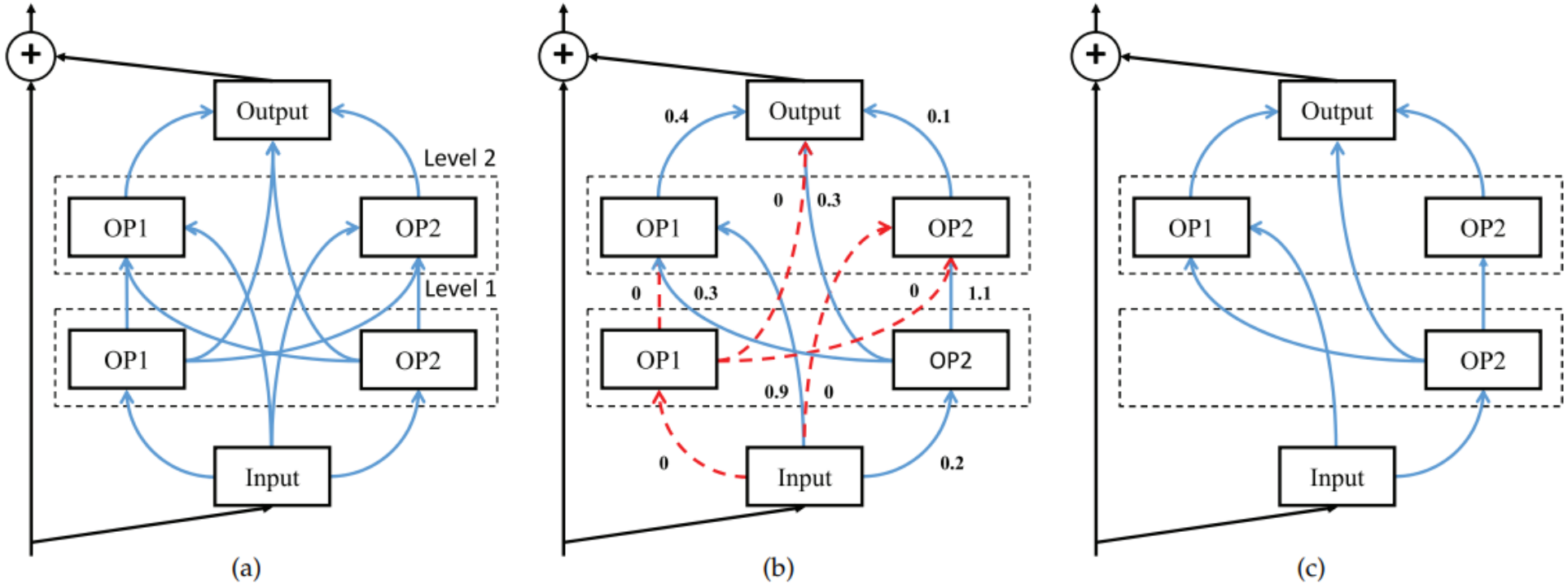


Fig. 5. An example of search block, which has two levels with two operations: (a) We start with a completely connected block in which all connections, and all nodes are kept. (b) In the search process, we jointly optimize the weights of neural network and the structure parameters λ (shown by the value on every connection) associated with each edge. (c) The final model after removing useless connections and operations.

Search Space

- Normal Setting

$$\mathbf{h}_{(b,i,j)} = \mathcal{O}_{(b,i,j)} \left(\sum_{m=1}^{i-1} \sum_{n=1}^{N} \mathbf{h}_{(b,m,n)} + \mathbf{O}_{(b-1)} \right)$$

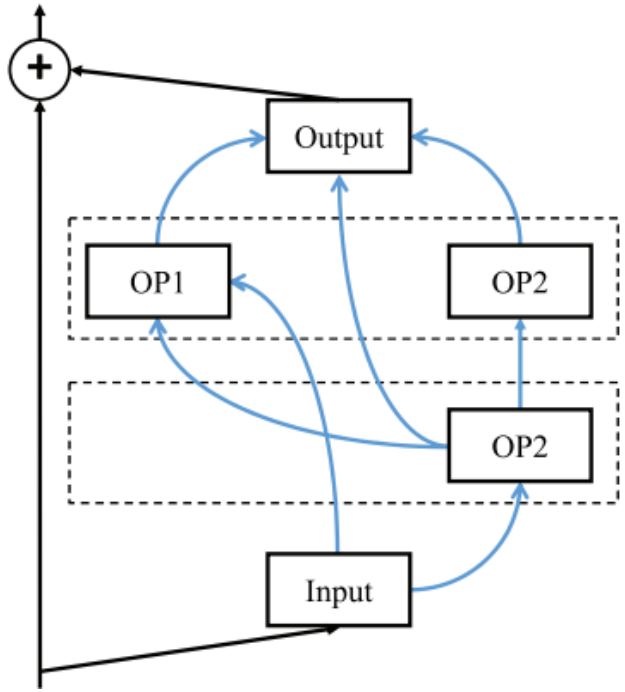
Conv 3x3

Conv 5x5

Conv 7x7

Max Pooling

Avg Pooling



An example of search block

- Mobile Setting

$$\mathbf{h}_{(b,i)} = \mathcal{O}_{(b,i)} \left(\sum_{i=1}^{N} \mathbf{h}_{(b,i)} + \mathbf{O}_{(b-1)} \right)$$

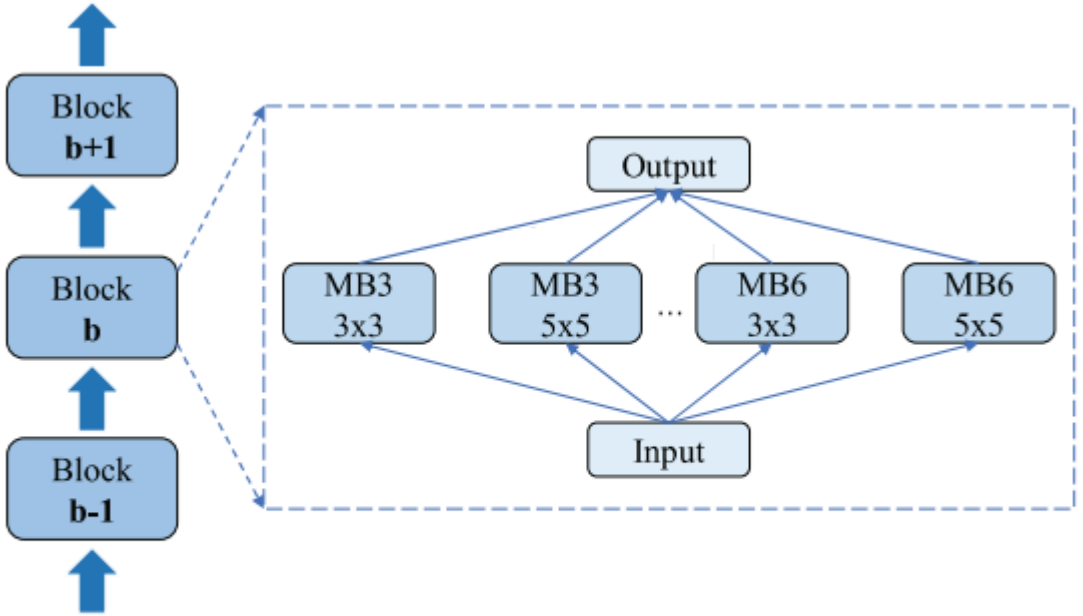


Fig. 3. The search space under mobile setting, where six operations are applied, namely, MBconv block with kernel sizes 3×3, 5×5, 7×7, expansion ratios with 3 and 6.

Training Loss

$$\min_{\boldsymbol{\lambda}} \ \mathcal{L}_{val}(\mathcal{F}(\boldsymbol{\lambda}, \mathbf{W}^*_{\boldsymbol{\lambda}})),$$
$$s.t.\ \mathbf{W}^*_{\boldsymbol{\lambda}} = \arg\min_{\mathbf{W}_{\boldsymbol{\lambda}}} \mathcal{L}_{train}(\mathcal{F}(\boldsymbol{\lambda}, \mathbf{W}_{\boldsymbol{\lambda}}))$$

- Structure Parameters $\boldsymbol{\lambda}$
- Network Weights $\boldsymbol{W}^*_{\boldsymbol{\lambda}}$

Optimization

$$\mathbf{z}_{(t)} = \boldsymbol{\lambda}'_{(t-1)} - \eta_{(t)} \nabla \mathcal{G}(\boldsymbol{\lambda}'_{(t-1)}) \quad (14)$$
$$\mathbf{v}_{(t)} = S_{\eta_{(t)}\gamma}(\mathbf{z}_{(t)}) - \boldsymbol{\lambda}'_{(t-1)} + \mu_{(t-1)}\mathbf{v}_{(t-1)} \quad (15)$$
$$\boldsymbol{\lambda}'_{(t)} = \mathcal{S}_{\eta_{(t)}\gamma}(\mathbf{z}_{(t)}) + \mu_{(t)}\mathbf{v}_{(t)}. \quad (16)$$

$\boldsymbol{\eta_t}$: $the\ gradient\ step\ size\ at\ iteration\ \boldsymbol{t}$

$S_{\boldsymbol{\eta}_{(t)\gamma}}$: $soft - threshold\ operator$

$$\mu_{(t-1)} = \frac{\boldsymbol{t-2}}{\boldsymbol{t+1}}$$

DSO-NAS Algorithm

Require: Training data D

1: Split training data D into two parts D_w and D_λ;
2: Initialize $\mathbf{W}_{\boldsymbol{\lambda}}$ and $\boldsymbol{\lambda}$;
3: Training $\mathbf{W}_{\boldsymbol{\lambda}}$ for several epochs on D_w;
4: **repeat**
5: Update $\boldsymbol{\lambda}^t$ by optimizing loss on dataset D_λ, $\mathcal{L}_{D_\lambda}(\mathbf{W}_{\boldsymbol{\lambda}^{t-1}}, \boldsymbol{\lambda}^{t-1})$ based on Eq. 14 to Eq. 16;
6: Update $\mathbf{W}_{\boldsymbol{\lambda}^t}$ by optimizing loss on dataset D_w, $\mathcal{L}_{D_w}(\mathbf{W}_{\boldsymbol{\lambda}^t}, \boldsymbol{\lambda}^{t-1})$ with NAG;
7: **until** converged

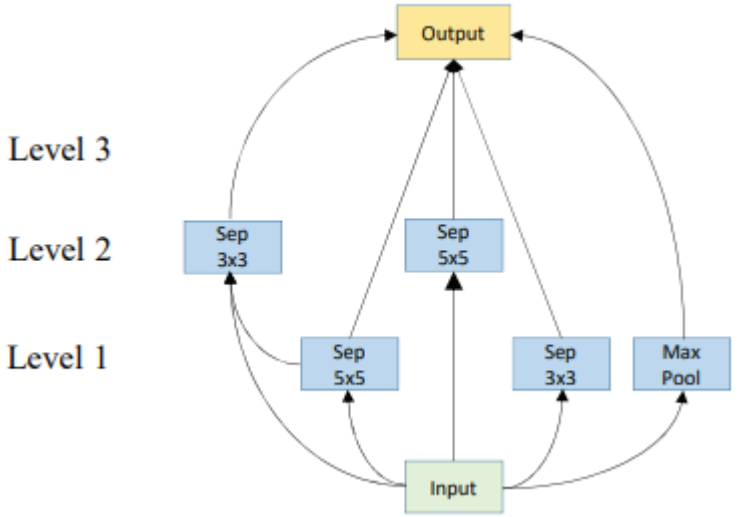
You Only Search Once: Single Shot Neural Architecture Search via Direct Sparse Optimization

Experiment Results

Dataset: CIFAR- 10

TABLE 1
Comparison with state-of-the-art NAS methods on CIFAR-10.

Architecture	Test Error	Parameter	Search Cost (GPU days)	Number of operations	Search Method	NAS Pipeline Completeness
ResNet(pre-activation) [63]	4.62	10.2M	-	-	manual	-
DenseNet [64]	3.46	25.6M	-	-	manual	-
NAS v1 [10]	5.50	4.2M	22400	-	RL	complete
NAS v2 [10]	6.01	2.5M	22400	-	RL	complete
NAS v3 [10]	4.47	7.1M	22400	-	RL	complete
NASNet-A [11]+cutout	2.65	3.3M	1800	13	RL	complete
NASNet-B [11]	3.73	2.6M	1800	13	RL	complete
AmoebaNet-A [9]	3.34	3.2M	3150	19	evolution	complete
AmoebaNet-B [9]+cutout	2.55	2.8M	3150	19	evolution	complete
PNAS [16]	3.41	3.2M	150	8	SMBO	complete
ENAS [12]+cutout	2.89	4.6M	0.5	5	RL	complete
Hierarchical Evo [65]	3.75	15.7M	300	6	evolution	complete
DARTS(1st order) [14]+cutout	2.94	2.9M	1.5	7	gradient-based	incomplete
DARTS(2nd order) [14]+cutout	2.83	3.4M	4	7	gradient-based	incomplete
SNAS(mild constraint) [52]+cutout	2.98	2.9M	1.5	7	gradient-based	complete
SNAS(moderate constraint) [52]+cutout	2.85 ± 0.02	2.8M	1.5	7	gradient-based	complete
SNAS(aggressive constraint) [52]+cutout	3.10 ± 0.04	2.3M	1.5	7	gradient-based	complete
GDAS [66]+cutout*	2.96	3.4M	0.2	7	gradient-based	complete
DARTS+ [49]+cutout*	2.67	3.7M	0.5	7	gradient-based	incomplete
DSO-NAS-share+cutout	2.74 ± 0.07	3.0M	1	4	gradient-based	complete
DSO-NAS-full+cutout	2.83 ± 0.12	3.0M	1	4	gradient-based	complete
random-share+cutout	3.48 ± 0.21	3.4 ± 0.1M	-	4	-	-
random-full+cutout	3.46 ± 0.19	3.5 ± 0.1M	-	4	-	-
full network+cutout	3.32 ± 0.05	3.0M	-	4	-	-



The block learned on CIFAR-10

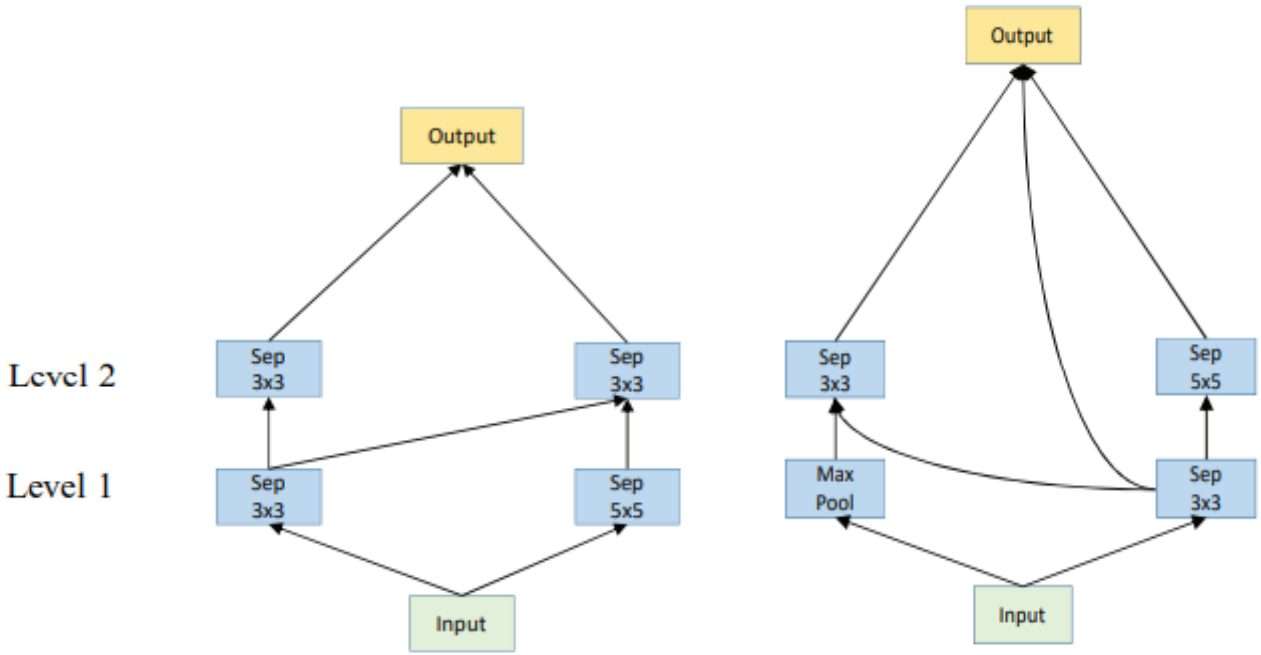
You Only Search Once: Single Shot Neural Architecture Search via Direct Sparse Optimization

Experiment Results

Dataset: ImageNet (ILSVRC 2012)

TABLE 2
Comparison with state-of-the-art image classifiers on ImageNet.

Architecture	Top-1/5 Error rate	Para	FLOPs	Search Cost (GPU days)
Inception-v1 [2]	30.2 /10.1	6.6M	1448M	-
MobileNet [76]	29.4 /10.5	4.2M	569M	-
ShuffleNet-v1 2x [77]	26.3 /10.2	5M	524M	-
ShuffleNet-v2 2x [54]	25.1 /-	5M	591M	-
NASNet-A† [11]	26.0 / 8.4	5.3M	564M	1800
NASNet-B† [11]	27.2 / 8.7	5.3M	488M	1800
NASNet-C† [11]	27.5 / 9.0	4.9M	558M	1800
AmoebaNet-A† [9]	25.5 / 8.0	5.1M	555M	3150
AmoebaNet-B† [9]	26.0 / 8.5	5.3M	555M	3150
AmoebaNet-C† [9]	24.3 / 7.6	6.4M	570M	3150
PNAS† [16]	25.8 / 8.1	5.1M	588M	150
DARTS† [14]	26.9 / 9.0	4.9M	595M	4
OSNAS [44]	25.8 /-	5.1M	-	-
MNAS-92 [45]	25.2 / 8.0	4.4M	388M	1600
GDAS† [66]	26.0 / 8.5	5.3M	581M	0.2
DARTS+ [49]*	25.3 / 8.1	5.1M	582M	6.8
DSO-NAS†	26.2 / 8.6	4.7M	571M	1
DSO-NAS-full	25.7 / 8.1	4.6M	608M	6
DSO-NAS-share	25.4 / 8.3	4.7M	567M	6
random-full	26.6 / 9.8	4.3M	583M	-
random-share	26.5 / 9.7	4.7M	564M	-



The blocks learned on ImageNet

You Only Search Once: Single Shot Neural Architecture Search via Direct Sparse Optimization

Experiment Results

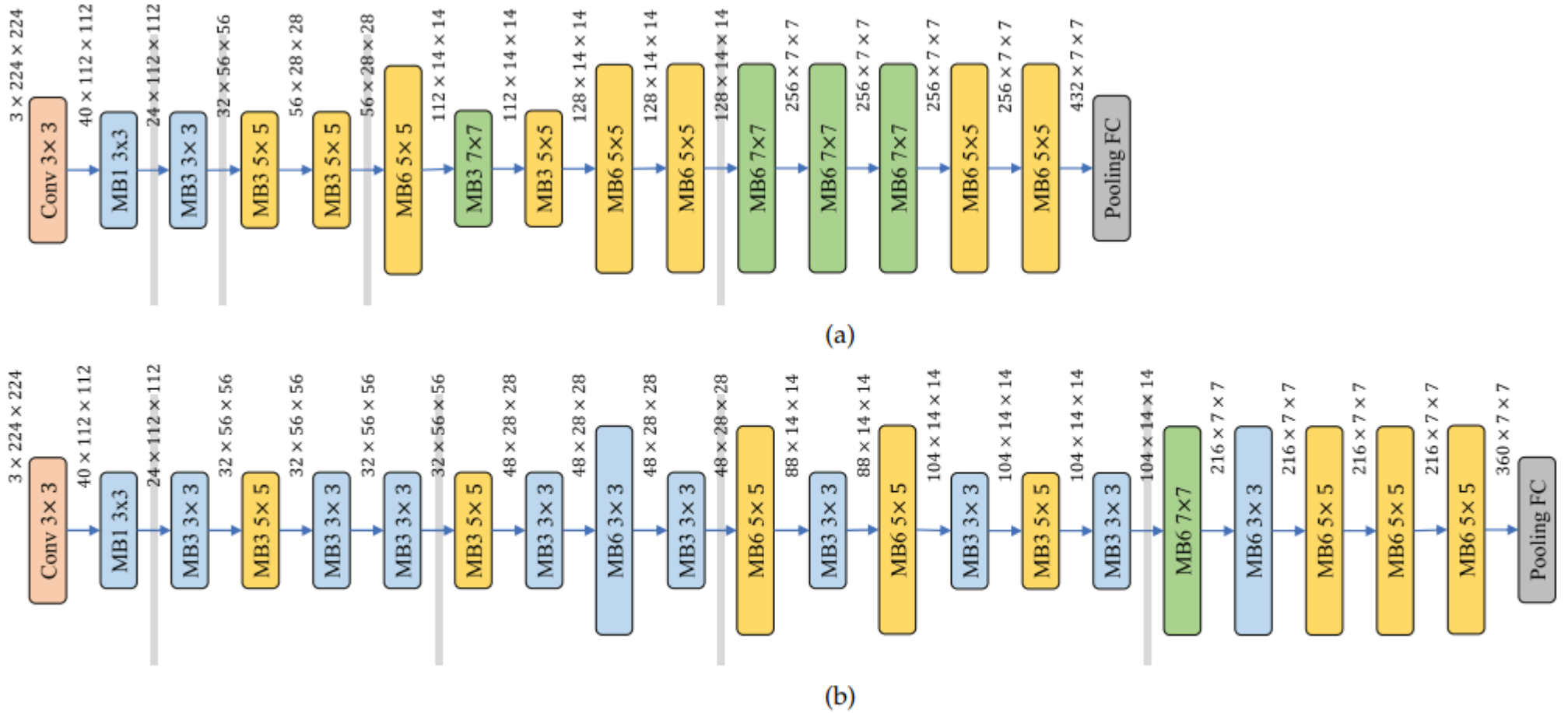


Fig. 7. Network structure learned on the search space proposed by Proxyless NAS. (a) Efficient GPU architecture obtained by DSO-NAS. (b) Efficient CPU architecture obtained by DSO-NAS.