Can weight sharing outperform random architecture search? An investigation with TuNAS

报告人:

时间：2020年7月2日

Outline

- Motivation
- Search space
- Method
- Experiments

motivation

Problem:

efficient architecture search methods did not always outperform random search baselines.

implicit costs: MnasNet and ProxylessNAS require us to run multiple searches with different hyper-parameters to match a given latency target.

Reason:

1) existing algorithms are challenging to implement and tune

2) most negative results focus on fairly small datasets such as CIFAR-10 or PTB, and some are obtained on heavily restricted search spaces

Outline

- Motivation
- Search space
- Method
- Experiments

Search Space

Search Space	Built Around	Base Filter Sizes	Typical Choices within an Inverted Bottleneck Layer			
		(c_i's for each layer)	Expansion Ratio	Kernel	Output filter size	SE
ProxylessNAS	MobileNetV2	ProxylessNAS [5]	$\{3, 6\}$	$\{3, 5, 7\}$	c_i	✗
ProxylessNAS-Enlarged	MobileNetV2	$\times 2$ when stride $= 2$	$\{3, 6\}$	$\{3, 5, 7\}$	$c_i \times \{\frac{1}{2}, \frac{5}{8}, \frac{3}{4}, 1, \frac{5}{4}, \frac{3}{2}, 2\}$	✗
MobileNetV3-Like	MobileNetV3	$\times 2$ when stride $= 2$	$\{1, 2, 3, 4, 5, 6\}$	$\{3, 5, 7\}$	$c_i \times \{\frac{1}{2}, \frac{5}{8}, \frac{3}{4}, 1, \frac{5}{4}, \frac{3}{2}, 2\}$	{✗, ✓}

Table 2: Search spaces we use to evaluate our method. The first two are built around MobileNetV2, whereas the third search space uses the combination of ReLU and Swish activations and the new model head from MobileNetV3. We use a target inference time of 84ms for the first two (to compare against ProxylessNAS and MnasNet) and 57ms for the third search space (to compare against MobileNetV3).

Outline

- Motivation
- Search space
- Method
- Experiments

Method

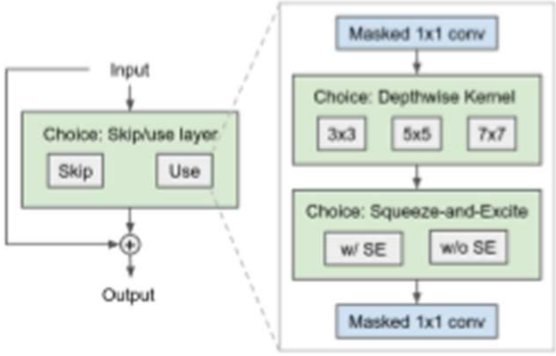
REINFORCE:

1) sample a network architecture $\alpha \sim \pi$.
2) we use the shared weights to estimate the quality $Q(\alpha)$ of the sampled architecture using a single batch of examples from the validation set.
   We then estimate the inference time of the sampled architecture $T(\alpha)$.
3) $Q(\alpha)$ and $T(\alpha)$ jointly determine the reward $r(\alpha)$,which is used to update the policy π using REINFORCE
4) we update the shared model weights W by computing a gradient update

Method

Weight sharing:
1 operation collapsing:

Search Space	Path number
Proxyless NAS	3x2=6
MobileNetV3-Like	6×3×7×2 =252



2 Channel Masking:
create only a single convolutional kernel with the largest possible number of channels. We simulate smaller channel sizes by retaining only the first N input (or output) channels, and zeroing out the remaining ones

Method

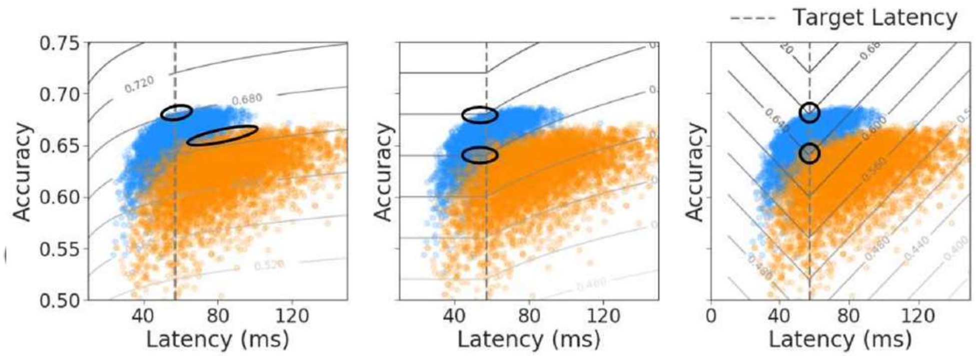
Reward Function

Soft Exponential Reward Function：

$$r(\alpha) = Q(\alpha) \times (T(\alpha)/T_0)^\beta$$

Hard Exponential Reward Function：

$$r(\alpha) = \begin{cases} Q(\alpha), & \text{if } T(\alpha) \le T_0 \\ Q(\alpha) \times (T(\alpha)/T_0)^\beta, & \text{if } T(\alpha) > T_0 \end{cases}$$



Absolute Reward Function

$$r(\alpha) = Q(\alpha) + \beta\,|T(\alpha)/T_0 - 1|$$

Method

Filter Warmup:

we randomly enable all the output filters – rather than just the filters selected by the RL controller – with some probability p.

we linearly decrease p from 1 to 0 over the first 25% of the search.

Op Warmup:

With some probability p between 0 and 1 we enable all operations within a choice block(rather than just enabling the operations selected by the RL controller)

When multiple operations are enabled, we average their outputs

we linearly decrease p from 1 to 0 over the first 25% of the search

Outline

- Motivation
- Search space
- Method
- Experiments

Experiment

Reproducing Reference Architectures

Name	Simulated Latency	Accuracy (%) Valid ours	Test published	Test ours
MobileNetV2	77.2 ms	74.5 ± 0.1	72.0	$\mathbf{73.3 \pm 0.1}$
MnasNet-B1	84.5 ms	76.0 ± 0.1	**74.5**	$\mathbf{74.5 \pm 0.1}$
ProxylessNAS	84.4 ms	76.3 ± 0.2	74.6	$\mathbf{74.9 \pm 0.1}$
MobileNetV3	58.5 ms	76.5 ± 0.2	75.2	$\mathbf{75.3 \pm 0.1}$

Table 3: Reproductions of our baseline models on ImageNet.

Experiment

Compare with random sample

Model / Method	Valid Acc (%)	Test Acc (%)	Latency
ProxylessNAS [5]	76.2	74.8	84.4
RS ($N = 20$)	75.4 ± 0.2	73.9 ± 0.3	84.3 ± 0.8
RS ($N = 50$)	75.4 ± 0.2	74.0 ± 0.2	83.8 ± 0.6
TuNAS (90 epochs)	76.3 ± 0.2	$\mathbf{75.0 \pm 0.1}$	84.0 ± 0.4

Model / Method	Valid Acc (%)	Test Acc (%)	Latency
MobileNetV2 [33]	74.4	73.4	77.2
MNASNet-B1 [36]	76.0	74.5	84.5
ProxylessNAS [5]	76.2	74.8	84.4
RS ($N = 20$)	74.4 ± 0.5	73.1 ± 0.6	84.0 ± 0.6
RS ($N = 50$)	74.6 ± 0.3	73.2 ± 0.3	83.5 ± 0.3
TuNAS (90 epochs)	76.4 ± 0.1	$\mathbf{75.3 \pm 0.2}$	84.0 ± 0.4

Model / Method	Valid Acc (%)	Test Acc (%)	Latency
MobileNetV3-L [13]	76.5	75.3	58.5
RS ($N = 20$)	74.1 ± 0.6	73.0 ± 0.5	58.5 ± 0.5
RS ($N = 50$)	74.6 ± 0.3	73.5 ± 0.2	58.7 ± 0.4
TuNAS (90 epochs)	76.6 ± 0.1	75.2 ± 0.2	57.0 ± 0.2
TuNAS (360 epochs)	76.7 ± 0.2	$\mathbf{75.4 \pm 0.1}$	57.1 ± 0.1

Experiment

object detection

Backbone	COCO Test-dev mAP	Latency
MobileNetV2	20.7	126
MNASNet	21.3	129
ProxylessNAS	21.8	140
MobileNetV3-Large	22.0	106
TuNAS Search	**22.5**	106

Table 7: Backbone architecture search results on MS COCO in the MobileNetV3-Like space. All detection backbones are combined with the SSDLite head. Target latency for TuNAS search was set to 106ms (same as for MobileNetV3-Large + SSDLite).

Experiment

Output filter sizes are important

Filters	Valid Acc (%)	Test Acc (%)	Latency
ProxylessNAS	76.3 ± 0.2	$\mathbf{75.0 \pm 0.1}$	84.0 ± 0.4
$\times 2$ Every Stride-2	74.8 ± 0.2	73.5 ± 0.2	83.9 ± 1.0
$\times 2$ Every Block	75.3 ± 0.2	74.0 ± 0.2	83.9 ± 0.2

Table 8: Effect of output filter sizes on final model accuracies.

Experiment

absolute value reward reduces

Reward	Valid Acc (%)	Test Acc (%)	Latency
MnasNet-Soft Reward	76.2 ± 0.2	74.8 ± 0.3	79.5 ± 3.3
Absolute Value Reward	76.4 ± 0.1	75.0 ± 0.1	84.1 ± 0.4

$\beta \in \{-0.03, -0.04, -0.05, -0.06, -0.07, -0.08, -0.09\}$

Warm up

Search Space	Warmup	Valid Acc (%)	Test Acc (%)	Latency
ProxylessNAS	✗	76.1 ± 0.1	74.7 ± 0.1	84.0 ± 0.3
ProxylessNAS	✓	76.3 ± 0.2	$\mathbf{75.0} \pm \mathbf{0.1}$	84.0 ± 0.4
ProxylessNAS-Enl	✗	75.8 ± 0.3	74.6 ± 0.2	83.6 ± 0.2
ProxylessNAS-Enl	✓	76.4 ± 0.1	$\mathbf{75.3} \pm \mathbf{0.2}$	84.0 ± 0.4
MobileNetV3-Like	✗	76.2 ± 0.2	75.0 ± 0.1	57.0 ± 0.6
MobileNetV3-Like	✓	76.6 ± 0.1	$\mathbf{75.2} \pm \mathbf{0.2}$	57.0 ± 0.2

The End!