DSNAS: Direct Neural Architecture Search without Parameter Retraining

报告人：时间：2020年7月9日

Outline

- Motivation
- Search space
- Method
- Experiments

motivation

Problem:

Most of the NAS methods proposed recently are implicitly two-stage methods. These two stages are searching and evaluation.

The correlation between the performance at the end of searching and aftert here training in evaluation is fairly low.

Contribution:

propose a plug-and-play NAS framework, DSNAS, as an efficient solution to this problem in large scale.

Outline

- Motivation
- Search space
- Method
- Experiments

Search Space

Input	Block	Channels	Repeat	Stride
$224^2 \times 3$	3×3 Conv	16	1	2
$112^2 \times 16$	CB	64	4	2
$56^2 \times 64$	CB	160	4	2
$28^2 \times 160$	CB	320	8	2
$14^2 \times 320$	CB	640	4	2
$7^2 \times 640$	1×1 Conv	1024	1	1
$7^2 \times 1024$	GAP	-	1	-
1024	FC	1000	1	-

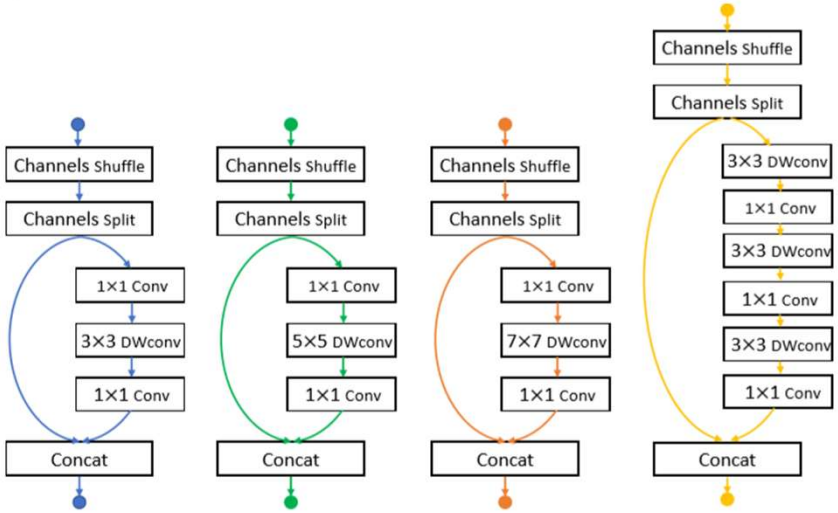


Figure 4. Choice blocks with stride=1. Choice blocks in search space. From left to right: Choice_3, Choice_5, Choice_7, Choice_x.

Outline

- Motivation
- Search space
- Method
- Experiments

Method

$$\tilde{\boldsymbol{O}}_{i,j}(\cdot) = \boldsymbol{Z}_{i,j}^{T} \boldsymbol{O}_{i,j}(\cdot). \quad (1)$$

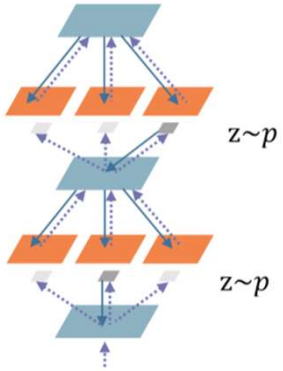
Forward，Z is random variables

$$\mathbb{E}_{\boldsymbol{Z} \sim p_{\boldsymbol{\alpha}}(\boldsymbol{Z})}[L_{\boldsymbol{\theta}}(\boldsymbol{Z})], \quad (2)$$

Monte Carlo estimate of the expectation of task objecteve Lθ(Z)

$$\frac{\partial \mathcal{L}}{\partial \alpha_{i,j}^{k}} = \frac{\partial \mathcal{L}}{\partial x_j} \boldsymbol{O}_{i,j}^{T}(x_i)(\delta(k' - k) - \tilde{\boldsymbol{Z}}_{i,j}) Z_{i,j}^{k} \frac{1}{\lambda \alpha_{i,j}^{k}}, \quad (3)$$

Gradient，λ is temperature



SNAS

Method

$$\mathbb{E}_{\tilde{\boldsymbol{Z}}\sim p(\tilde{\boldsymbol{Z}})}[\frac{\partial \mathcal{L}}{\partial \alpha_{i,j}^k}]$$
$$= \mathbb{E}_{\tilde{\boldsymbol{Z}}\sim p(\tilde{\boldsymbol{Z}})}[\nabla_{\alpha_{i,j}^k} \log p(\tilde{\boldsymbol{Z}})[\frac{\partial \mathcal{L}}{\partial x_j} \boldsymbol{O}_{i,j}^T(x_i)\tilde{\boldsymbol{Z}}_{i,j}]_c], \tag{4}$$

Policy gradient

$$\lim_{\lambda\to 0} \mathbb{E}_{\tilde{\boldsymbol{Z}}\sim p(\tilde{\boldsymbol{Z}})}[\frac{\partial \mathcal{L}}{\partial \alpha_{i,j}^k}]$$
$$= \lim_{\lambda\to 0} \mathbb{E}_{\tilde{\boldsymbol{Z}}\sim p(\tilde{\boldsymbol{Z}})}[\nabla_{\alpha_{i,j}^k} \log p(\tilde{\boldsymbol{Z}}_{i,j})[\frac{\partial \mathcal{L}}{\partial x_j} \tilde{\boldsymbol{O}}_{i,j}(x_i)]_c]$$
$$= \mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z})}[\nabla_{\alpha_{i,j}^k} \log p(\boldsymbol{Z}_{i,j})[\frac{\partial L}{\partial x_j} \boldsymbol{O}_{i,j}^T(x_i)\boldsymbol{Z}_{i,j}]_c]$$
$$= \mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z})}[\nabla_{\alpha_{i,j}^k} \log p(\boldsymbol{Z}_{i,j})[\frac{\partial L}{\partial x_j} \sum_k O_{i,j}^k(x_i)Z_{i,j}^k]_c], \tag{5}$$

Method

$$\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z})}[\nabla_{\alpha_{i,j}^k} \log p(\boldsymbol{Z}_{i,j})[\frac{\partial L}{\partial x_j}\sum_k O_{i,j}^k(x_i)Z_{i,j}^k]_c], \tag{5}$$

$$\begin{aligned} C(\boldsymbol{Z}_{i,j}) &= \sum_k \frac{\partial L}{\partial x_j} O_{i,j}^k(x_i) Z_{i,j}^k \\ &= \frac{\partial L}{\partial x_j^i} O_{i,j}^s(x_i) Z_{i,j}^s = \frac{\partial L}{\partial x_j^i} x_j^i \\ &= \frac{\partial L}{\partial x_j^i}\frac{\partial x_j^i}{\partial Z_{i,j}^s} = \frac{\partial L}{\partial Z_{i,j}^s}, \end{aligned} \tag{6}$$

$x_j^i = O_{i,j}^s(x_i)Z_{i,j}^s$ is the output of the operation $O_{i,j}^s$ Chosen at edge(i,j)

Method

Algorithm 1 Discrete SNAS

Require: parent network, operation parameters $\boldsymbol{\theta}$ and categorical arch distribution $p_{\boldsymbol{\alpha}}(\boldsymbol{Z})$

Initialize $\boldsymbol{\theta}$, $\boldsymbol{\alpha}$

while not converged **do**

Sample one-hot random variables $\boldsymbol{Z}$ from $p_{\boldsymbol{\alpha}}(\boldsymbol{Z})$

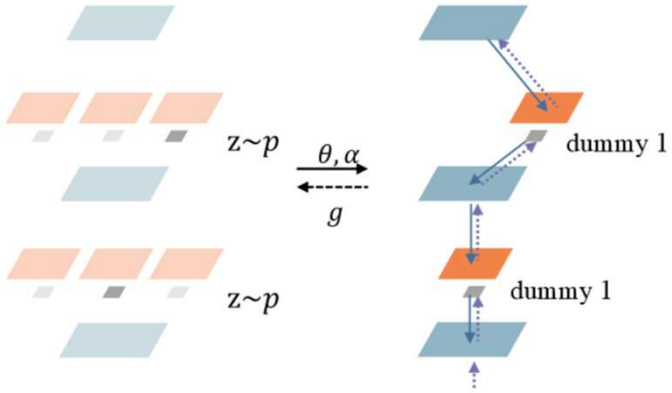
Construct child network with $\boldsymbol{\theta}$ according to $\boldsymbol{Z}$, multiply a 1^{dummy} after each feature map X

Get a batch from data and forward to get L

Backward L to both $\boldsymbol{\theta}$ and 1^{dummy}, backward $\log p_{\boldsymbol{\alpha}}(\boldsymbol{Z})$ to $\boldsymbol{\alpha}$

Update $\boldsymbol{\theta}$ with $\frac{\partial L}{\partial \boldsymbol{\theta}}$, update $\boldsymbol{\alpha}$ with $\frac{\partial \log p_{\boldsymbol{\alpha}}(\boldsymbol{Z})}{\partial \boldsymbol{\alpha}} \frac{\partial L}{\partial 1^{dummy}}$

end while



DSNAS

Method

Method	Forward time	Backward time	Memory
Subnetwork	$O(P)$	$O(Q)$	$O(M)$
SNAS	$O(nP)$	$O(nQ)$	$O(nM)$
ProxylessNAS*	$O(nP)$	$O(nQ)$	$O(nM)$
ProxylessNAS	$O(2P)$	$O(2Q)$	$O(2M)$
DSNAS	$O(P)$	$O(Q)$	$O(M)$

Table 1. Computation complexity comparison between SNAS, ProxylessNAS and DSNAS. ProxylessNAS* indicates its theoretical complexity.

Method

Progressive early stop:

The non-zero entropy keeps the sampling going on until the end, regardless of the fact that sampling at that uncertainty level does not bring significant gains.

$$\min\{\alpha_{i,j}^{k} - \alpha_{i,j}^{m}, \forall m \neq k || \alpha_{i,j}^{k} = \max\{\boldsymbol{\alpha}_{i,j}\}\} \geq h. \quad (8)$$

Comparison with one-shot NAS:

one-shot NAS only do architecture optimization in one-shot, all of them are two-stage methods.

DSNAS optimizes architecture alongside with parameters, which is expected to save some resources in the pretraining stage.

superiority：

One-short methods can also do finetuning, but still parameters of the optimal networks are updated less frequently than DSNAS

Outline

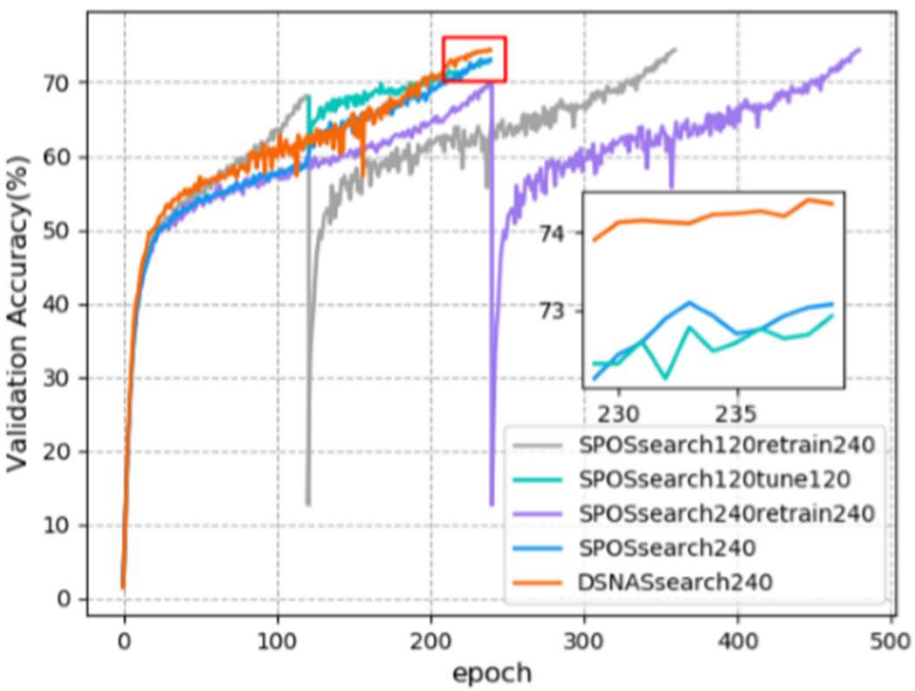
- Motivation
- Search space
- Method
- Experiments

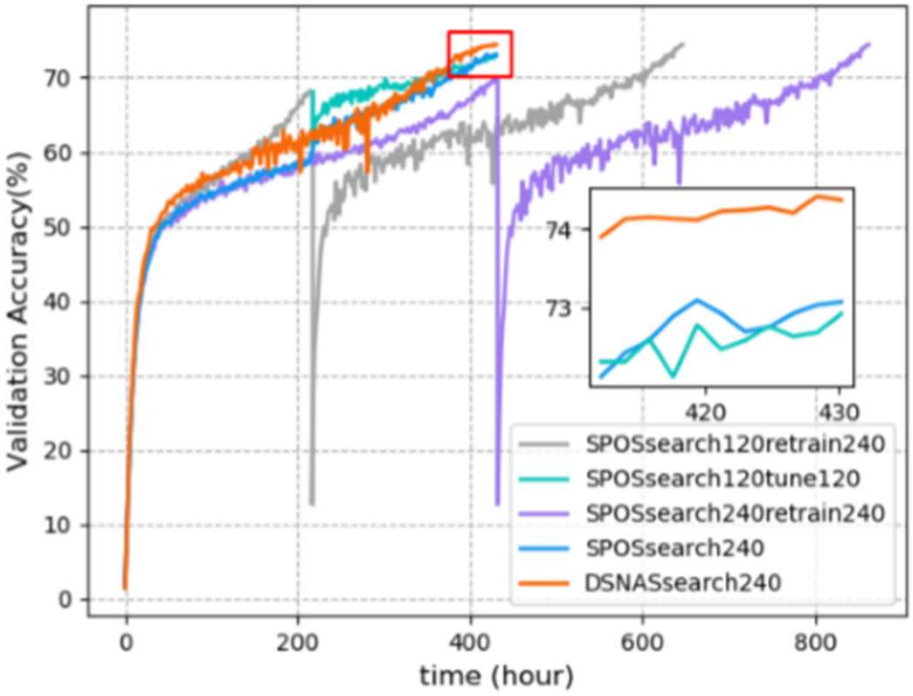
Experiment

$$\tau = \frac{2(N_{concordant} - N_{disconcordant})}{N(N-1)}, \qquad (9)$$

Model	τ_{inter}	τ_{intra}
Single Path One-Shot[7]	0.33	-0.07
ProxylessNas [3]	-	-0.33

Experiment


Validation Accuracy(%)
epoch
0
100
200
300
400
500
0
10
20
30
40
50
60
70
74
73
230
235
SPOSsearch120retrain240
SPOSsearch120tune120
SPOSsearch240retrain240
SPOSsearch240
DSNASsearch240


Validation Accuracy(%)
time (hour)
0
200
400
600
800
0
10
20
30
40
50
60
70
74
73
420
430
SPOSsearch120retrain240
SPOSsearch120tune120
SPOSsearch240retrain240
SPOSsearch240
DSNASsearch240

Experiment

Model	FLOPS	Search		Retrain		No proxy	Time (GPU hour)	
		Top-1 acc(%)	Top-5 acc(%)	Top-1 acc(%)	Top-5 acc(%)		Search	Retrain
MobileNet V1 (0.75x)[8]	325M	Manual		68.4	-		Manual	
MobileNet V2 (1.0x)[19]	300M	Manual		72.0	91.00		Manual	
ShuffleNet V2 (1.5x)[15]	299M	Manual		72.6	90.60		Manual	
NASNET-A(4@1056)[29]	564M	-	-	74.0	91.60	False	48000	-
PNASNET[13]	588M	-	-	74.2	91.90	False	5400	-
MnasNet-A1[23]	312M	-	-	75.2	92.50	False	40000§	-
DARTS[14]	574M	-	-	73.3	91.30	False	24	288
SNAS[27]	522M	-	-	72.7	90.80	False	36	288
Proxyless-R (mobile)[3]	320M	62.6*	84.7*	74.6	92.20	True	300‡	≥384
Single Path One-Shot[7]	319M	68.7†	-	74.3	-	True	250	384
Single Path One-Shot*	323M	68.2±0.38	88.28	74.3±0.36	91.79	True	250	384
Random Search	≤330M	≤68.2	≤88.31	≤73.9	≤91.8	True	250	384
DSNAS	324M	**74.4±0.22**	91.54	74.3±0.27	91.90	True	420	

Experiment

Method	Train			Test
	Forward	Backward	Opt	
SNAS	0.26s	0.46s	0.14s	0.18s
DSNAS	0.05s	0.07s	0.13s	0.04s

Table 4. Computation time of SNAS and DSNAS

Method	Train			Test
	Forward	Backward	Opt	
ProxylessNAS	3.3s	2.3s	3.6s	1.2s
DSNAS	1.9s	1.3s	2.6s	0.9s

Table 5. Computation time of ProxylessNAS and DSNAS

The End!