DBS: DIFFERENTIABLE BUDGET-AWARE SEARCHING FOR CHANNEL PRUNING

Zhaokai Zhang⋆ , Tianpeng Feng⋆ , Yang Liu, Chunnan Sheng, Fanyi Wang, He Cai

OPPO Research Institute

Motivation

Network pruning is an effective technique to reduce computation costs for deep model deployment on resource-constraint devices. Searching superior sub-networks from a vast search space through NAS, which conducts a one-shot supernet used as a performance estimator, is still time-consuming. In addition to searching ineffciency, such solutions also focus on FLOPs budget and suffer from an inferior ranking consistency between supernet-inherited and stand-alone performance.

Method

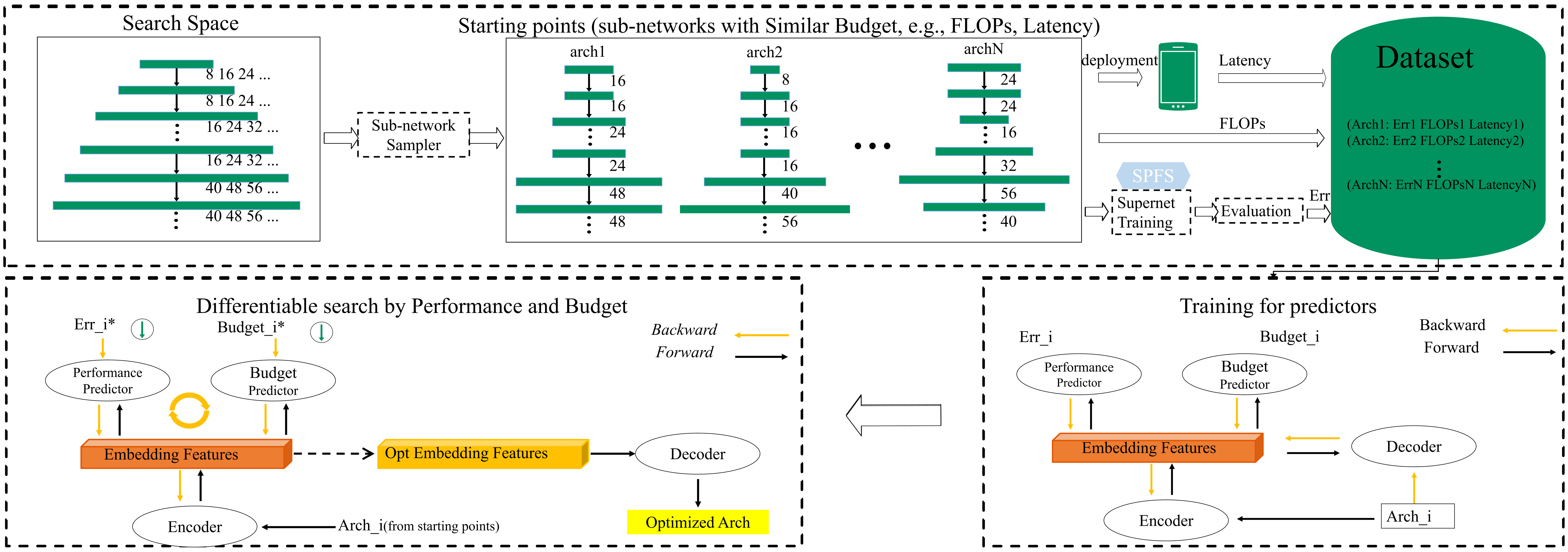

Fig. 1. The pipeline of DBS. 1) Train and evaluate a supernet relying on starting points; 2) Train Transformer-based predictors, including performance predictor and budget predictor; 3) Freeze the parameters of predictors and Perform a differentiable budget-aware search on embedding features; 4) Decode corresponding sub-networks from optimized features. SPFS: Strict Path-wise Fair Sandwich rule

Experment results

MobileNet-V2				Resnet-18			
Method	FLOPs	Top-1	△Top-1	Method	FLOPs	Top-1	△Top-1
Uniform 1.0x	300M	72.3	0	Uniform 1.0x	1.8G	70.1	0
MetaPruning	313M	72.7	0.4	DMCP	1.04G	69.2	-0.9
AutoSlim*	300M	74.2	1.9	Cafenet	1.2G	71.2	1.1
DMCP	300M	73.9	1.6	Cafenet	0.9G	70.8	0.7
DMCP*	300M	74.6	2.3	CHEX	1.04G	69.2	-0.9
DBS	**299M**	**74.3**	**2**	**DBS**	**1.04G**	**71.4**	**1.3**
DBS*	**299M**	**75**	**2.7**	**Resnet-50**			
Uniform 0.75x	210M	70.3	0	Uniform 0.85x	3.0G	75.3	0
AMC	211M	70.8	0.5	MetaPruning	3.0G	76.2	0.9
MetaPruning	217M	71.2	0.9	AutoSlim	3.0G	76	0.7
AutoSlim*	211M	73	2.7	DMCP	2.8G	77	1.7
DMCP	211M	72.4	2.1	Cafenet	3.0G	77.4	2.1
DMCP*	211M	73.5	3.2	**DBS**	**2.8G**	**78.2**	**2.9**
Cafenet	217M	73.4	3.1	Uniform 0.75x	2.3G	74.6	0
CHEX	220M	72	1.7	MetaPruning	2.3G	75.4	0.8
DBS	**212M**	**73.1**	**2.8**	AutoSlim	2.0G	75.6	1
DBS*	**212M**	**73.7**	**3.4**	DMCP	2.2G	76.2	1.6
Uniform 0.5x	97M	65.4	0	Cafenet	2.0G	76.9	2.3
DMCP	97M	67	1.6	CHEX	2.0G	77.4	2.8
Cafenet	106M	68.7	3.3	**DBS**	**2.2G**	**77.8**	**3.2**
DBS	**97M**	**68.2**	**2.8**	Uniform 0.5x	1.1G	71.9	0
MetaPruning	87M	63.8	0	MetaPruning	1.1G	73.4	1.5
DMCP	87M	66.1	2.3	AutoSlim	1.1G	74	2.1
DBS	**87M**	**66.6**	**2.8**	DMCP	1.1G	74.4	2.5
Uniform 0.35x	59M	60.1	0	Cafenet	1.0G	75.3	3.4
DMCP	59M	62.7	1.6	CHEX	1.0G	76	4.1
DBS	**59M**	**63**	**2.9**	**DBS**	**1.1G**	**76.5**	**4.6**
MetaPruning	43M	58.3	0	Uniform 0.25x	278M	63.5	0
DMCP	43M	59.1	0.8	DMCP	278M	68.3	4.8
DBS	**43M**	**60.5**	**2.2**	**DBS**	**278M**	**69**	**5.5**

Table 1. Results of pruned models from MobileNet-V2, Resnet-18 and Resnet-50 under various FLOPs settings. * indicates the pruned model is trained by the slimmable method.

Backbone	Method	Latency	FLOPs	Top-1	△Top-1
Resnet-18	uniform	21.96ms	1.04G	68.4	0
	DMCP	24.27ms	1.04G	69.2	0.8
	DBS†	**22.55ms**	**1.04G**	**69.5**	**1.1**
	DBS‡	**21.98ms**	**1.04G**	**69.5**	**1.1**
Resnet-50	uniform	84.40ms	2.80G	76.5	0
	DMCP	87.54ms	2.80G	77	0.5
	DBS†	**87.94ms**	**2.80G**	**77.2**	**0.7**
	DBS‡	**75.88ms**	**2.80G**	**77.3**	**0.8**
	uniform	67.66ms	2.20G	76.1	0
	DMCP	69.98ms	2.20G	76.2	0.1
	DBS†	**67.45ms**	**2.20G**	**76.5**	**0.4**
	DBS‡	**64.46ms**	**2.21G**	**76.6**	**0.5**
	uniform	37.25ms	1.10G	73.7	0
	DMCP	38.78ms	1.10G	74.4	0.7
	DBS†	**36.2ms**	**1.10G**	**74.8**	**1.1**
	DBS‡	**33.63ms**	**1.12G**	**74.8**	**1.1**
	uniform	12.25ms	278M	66.5	0
	DMCP	12.75ms	278M	68.3	1.8
	DBS†	**11.33ms**	**278M**	**68.6**	**2.1**
	DBS‡	**11.14ms**	**284M**	**68.4**	**1.9**

Table 2. Comparison of latency. † represents search results by FLOPs. ‡ means search results by latency.

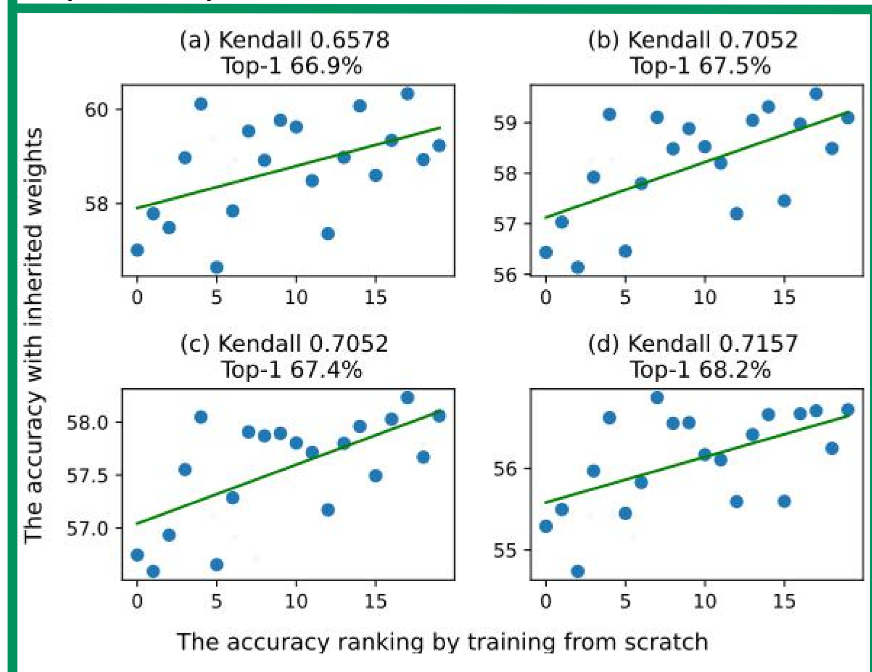

Fig. 2. Rank consistency visualization of different strategies. a: Uniform sampling, b: Sandwich rule, c: Strict path-wise fair rule, d: Strict path-wise fair sandwich rule.

Backbone	Search algorithm	FLOPs	Top-1	△Top-1
MBV2	random	300M	73.2	0
	evolutionary	300M	73.8	0.6
	our	**300M**	**74.2**	**1**
	random	211M	72.5	0
	evolutionary	211M	72.8	0.3
	our	**211M**	**73.1**	**0.6**
	random	97M	67.4	0
	evolutionary	97M	67.9	0.5
	our	**97M**	**68.1**	**0.7**

Table 3. Results of different search algorithms.For fair comparisons, we perform random and evolutionary searches at the same cost. We conduct experiments under three FLOPs settings, our method can always fnd sub-networks with better performance