GreedyNASv2: Greedier Search with a Greedy Path Filter

Tao Huang[1,2] Shan You[1,3*] Fei Wang[4] Chen Qian[1] Changshui Zhang[3] Xiaogang Wang[1] Chang Xu[2]

[1]SenseTime Research [2]School of Computer Science, Faculty of Engineering, The University of Sydney
[3]Department of Automation, Tsinghua University [4]University of Science and Technology of China
youshan@sensetime.com

Motivation

NAS with search space shrinkage:

1. Enhancing the discriminability of one-shot supernet by greedily filtering out those weak paths.
2. **Path-level shrinkage:** remove specific paths in the search space.
3. **Operation-level shrinkage:** remove operations of each layers.

Limitations:

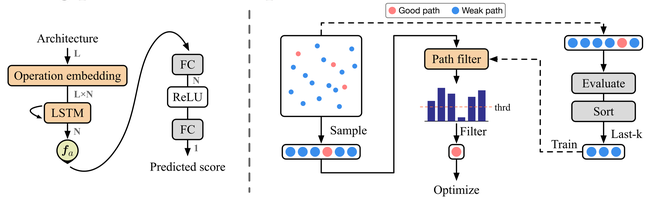
1. Too aggressive to train the elite paths with enough diversity with a limited number of evaluation paths.
2. Imprecise measuring of operation importance: manually-designed importance metrics, lack of intra-layer dependencies.

Intuition

1. Learn a path filter with the highly-confident weak paths. Therefore, the filter can locate the good and weak paths more precisely with limited evaluated paths.
2. Find unimportant operations using the learned path filter, which contains operation-level importance and intra-layer dependencies.

Greedier Sampling with a Path Filter

Learning path filter as PU prediction

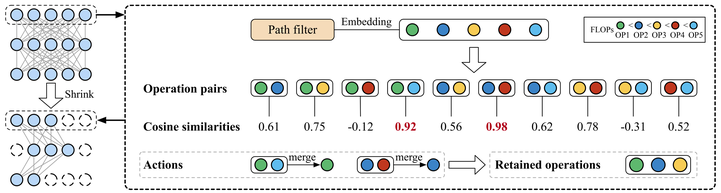


With the highly-confident "weak" paths evaluated by multi-path sampling in GreedyNAS, GreedyNASv2 proposes to leverage positive-unlabeled (PU) learning to train a promising path filter.

Supernet training:

1. Get the last-k out of m randomly-sampled paths by evaluating them on a small validation set.
2. Train the path filter using PU learning. Positive set: weak paths obtained in step 1; unlabeled set: random paths.
3. Use the path filter to filter those weak paths.

Shrinking operations with learned embeddings



Motivation:

- PU filter: distinguish whether a path is good or weak using the learned operation embeddings.
- If two operations of a layer have the same embedding, they will have the same influence on the classification results on every paths.
- $\longrightarrow$ we can merge the operations with similar embeddings and keep the less-costly ones.

Experiments

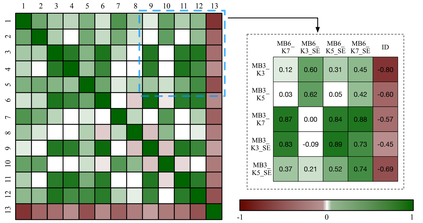
Experiments on different scales of search spaces [S-L]:

Method	ACC			supernet ACC		
	S	M	L	S	M	L
SPOS	76.8	76.6	75.5	56.5	48.2	33.4
GreedyNAS	77.1	76.8	76.5	57.6	49.3	35.1
GreedyNASv2	77.3	77.4	77.5	58.1	55.5	43.8

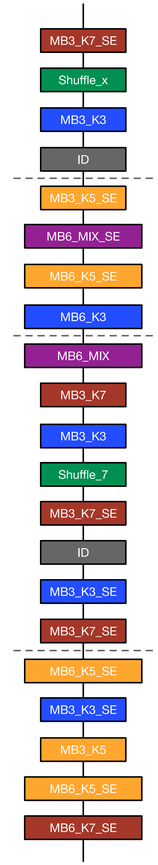
Results on the proposed ResNet search space:

Method	ACC (%)	FLOPs (M)	Params (M)	Cost (days)	Search num.
ResNet-50	76.1	4089	25.6	-	-
ResNeXt-50	77.8	4230	25.0	-	-
SE-ResNeXt-50	78.9	4233	27.6	-	-
SPOS	80.6	4153	27.8	15.4	1000
GreedyNAS	80.8	4125	28.1	11.3	1000
GreedyNASv2	81.1	4098	26.9	9.0	500

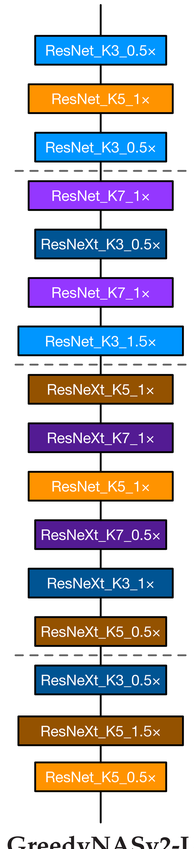
Visualization of the learned operation similarities:



Visualization of archs



GreedyNASv2-S



GreedyNASv2-L