GreedyNAS: Towards Fast One-Shot NAS with Greedy Supernet

Shan You[1,2*], Tao Huang[1,3*], Mingmin Yang[1*], Fei Wang[1], Chen Qian[1], Changshui Zhang[2]

[1]SenseTime [2]Department of Automation, Tsinghua University [3]Dian Group, School of CST, Huazhong University of Science and Technology *Equal contributions

{youshan,huangtao,yangmingmin,wangfei,qianchen}@sensetime.com zcs@mail.tsinghua.edu.cn

Motivation

One-shot NAS: Based on the weight-sharing paradigm, One-shot NAS methods model NAS as a one-shot training process of an over-parameterized supernet, where various architectures can be directly derived.

Supernet: matters as a fundamental performance estimator of different architectures (paths).

Target Assumption: The supernet should estimate the (relative) performance accurately for *all* paths, and thus all paths are treated equally and trained simultaneously.

Issues:

- It is harsh for a single supernet to evaluate accurately on such a huge-scale search space (*e.g.*, 7^{21}).
- There exist many architectures of inferior quality in terms of accuracy performance.
- Since the weights of all paths are highly shared, if a weak path is sampled and gets trained, it would disturb the weights of those potentially-good paths.
- Training on those weak paths actually involves unnecessary update of weights, and slows down the training efficiency more or less.

Intuition

- block the training of these weak paths.
- Consider a complete partition of search space $\mathcal{A}$ of two subsets $\mathcal{A}_{good}$ and $\mathcal{A}_{weak}$:

$$\mathcal{A} = \mathcal{A}_{good} \bigcup \mathcal{A}_{weak}, \ \mathcal{A}_{good} \bigcap \mathcal{A}_{weak} = \emptyset,$$

where for an Oracle supernet $\mathcal{N}_o$,

$$\mathrm{ACC}(\boldsymbol{a}, \mathcal{N}_o, \mathcal{D}_{val}) \geq \mathrm{ACC}(\boldsymbol{b}, \mathcal{N}_o, \mathcal{D}_{val})$$

holds for all $\boldsymbol{a} \in \mathcal{A}_{good}$, $\boldsymbol{b} \in \mathcal{A}_{weak}$ on validation dataset $\mathcal{D}_{val}$.

- Idea: just sample from the potentially-good paths $\mathcal{A}_{good}$ instead of all paths $\mathcal{A}$,

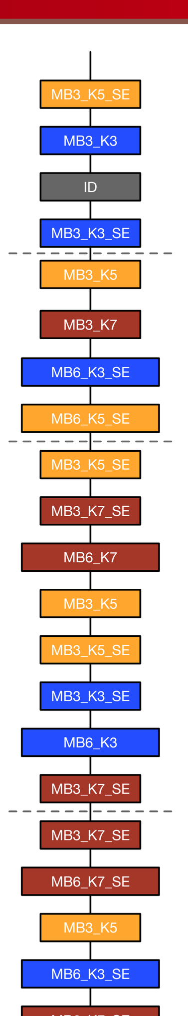
$$p(\boldsymbol{a}; \mathcal{N}_o, \mathcal{D}_{val}) = \frac{1}{|\mathcal{A}_{good}|} \mathbb{I}(\boldsymbol{a} \in \mathcal{A}_{good}).$$

Problems:

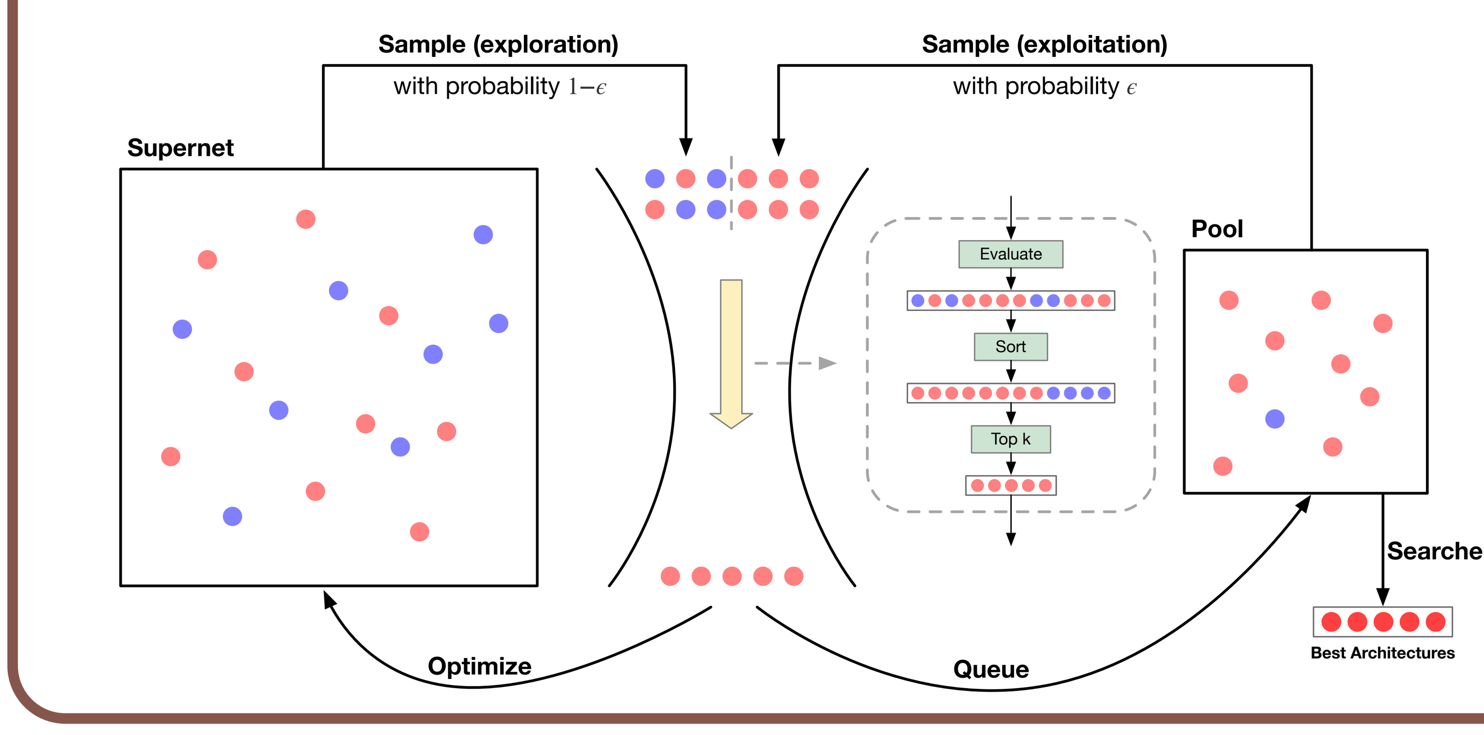
1. Q: Oracle supernet $\mathcal{N}_o$ is unknown.
   A: greedily use current supernet $\mathcal{N}_\dagger$ as a proxy
2. Q: How can we accurately identify whether a path is from $\mathcal{A}_{good}$ or $\mathcal{A}_{weak}$?
   A: multi-path sampling with rejection.

Experimental Settings

- Search space: MobileNetV2 inverted bottleneck with CNN kernel $\{3,5,7\}$ and expansion ratio $\{3,6\}$. Size 7^{21} with identity. Larger size 13^{21} with SE.
- ImageNet dataset: 50K validation, 50K testing
- Supernet: sample 10 paths and filter 5, 1K images for path filtering, pool size 1K, SGD optimizer
- Searching: evolutionary NSGA-II
- Retraining: following Proxyless-NAS without SE and Mnasnet with SE.



Framework of GreedyNAS



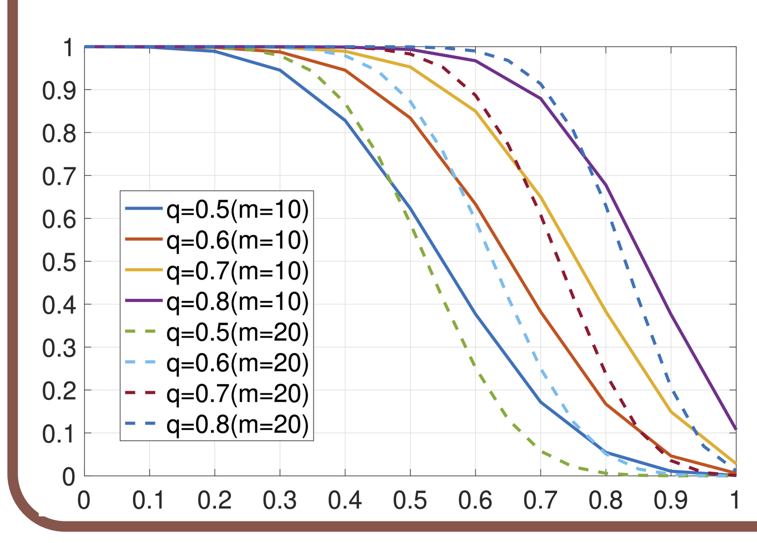
The supernet greedily shrinks its training space from all paths (red and blue dots) into potentially-good paths (red dots), and further into candidate pool.

Multi-path Sampling with Rejection

Theorem 1 *If m paths are sampled uniformly i.i.d. from $\mathcal{A}$, then it holds that at least k ($k \leq m$) paths are from $\mathcal{A}_{good}$ with probability*

$$\sum_{j=k}^{m} \mathrm{C}_m^j q^j (1-q)^{m-j}, \tag{1}$$

where $q = |\mathcal{A}_{good}| / |\mathcal{A}|$.



Eq.(1) can be large; with $q = 0.6$, it has 83.38% confidence to say at least 5 out of 10 paths are from $\mathcal{A}_{good}$.

Solution: just rank the sampled m paths using a small portion of validation data $\mathcal{D}_{val}$, keep the Top-k paths and reject the remaining paths.

Input: number of sampled multiple paths m, number of kept paths k, candidate pool $\mathcal{P}$

1: **if** without candidate pool $\mathcal{P}$ **then**
2: sample m paths $\{\boldsymbol{a}_i\}_{i=1}^m$ i.i.d. w.r.t. $\boldsymbol{a}_i \sim U(\mathcal{A})$
3: **else**
4: sample m paths $\{\boldsymbol{a}_i\}_{i=1}^m$ i.i.d. w.r.t. $\boldsymbol{a}_i \sim (1-\epsilon) \cdot U(\mathcal{A}) + \epsilon \cdot U(\mathcal{P})$
5: **end if**
6: randomly sample a batch $\hat{\mathcal{D}}_{val}$ in $\mathcal{D}_{val}$
7: evaluate the loss ℓ_i of each path $\boldsymbol{a}_i$ on $\hat{\mathcal{D}}_{val}$
8: rank the paths by ℓ_i, and get Top-k indexes $\{t_i\}_{i=1}^k$
9: return k paths $\{\boldsymbol{a}_{t_i}\}_{i=1}^k$ and filter the rest

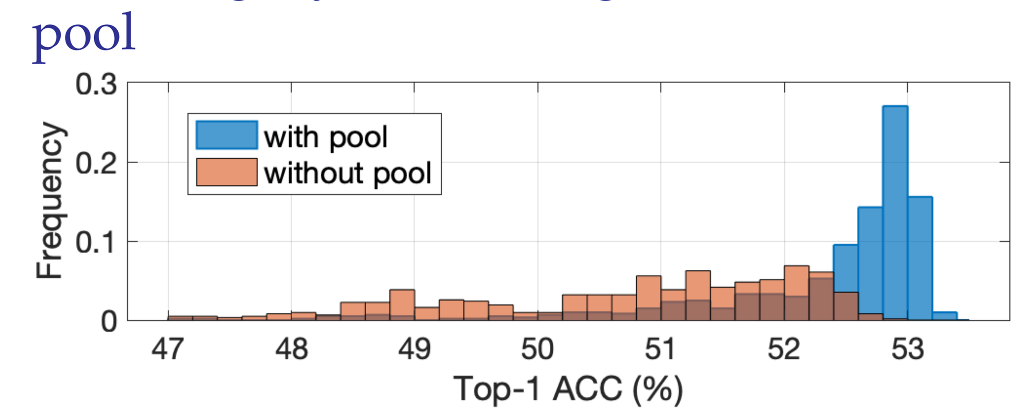
Exploration and Exploitation Training with Candidate Path Pool

We introduce a candidate path pool to store the discovered good paths, and sample from it,

$$\boldsymbol{a} \sim (1-\epsilon) \cdot U(\mathcal{A}) + \epsilon \cdot U(\mathcal{P}), \tag{2}$$

Four advantages:

1. boosting the training efficiency
2. increasing the probability of sampling good paths $q = \epsilon + (1-\epsilon)|\mathcal{A}_{good}|/|\mathcal{A}|$, *e.g.* from 83.38% to 99.36% for 5/10 with $\epsilon = 0.5$
3. stopping principle via candidate pool
   Stop by observing the steadiness of pool $\pi := \frac{|\mathcal{P}_t \bigcap \mathcal{P}|}{|\mathcal{P}|} \leq \alpha$
4. searching by initializing with candidate pool



Searching Results with Same Search Space on ImageNet

Methods	performance			supernet training efficiency		
	Top-1 (%)	FLOPs	latency	#optimization	#evaluation	corrected #optimization
Proxyless-R (mobile)	74.60	320M	79 ms	-	-	-
Random Search	74.07	321M	69 ms	1.23M×120	-	147.6M
Uniform Sampling	74.50	326M	72 ms	1.23M×120	-	147.6M
FairNAS-C	74.69	321M	75 ms	1.23M×150	-	184.5M
Random Search-E	73.88	320M	91 ms	1.23M×73	-	89.8M
Uniform Sampling-E	74.17	320M	94 ms	1.23M×73	-	89.8M
GreedyNAS	**74.85**	320M	89 ms	1.23M×46	2.40M×46	89.7M
GreedyNAS	**74.93**	324M	78 ms	1.23M×46	2.40M×46	89.7M

Comparison with State-of-the-art NAS Methods on ImageNet

Methods	Top-1 (%)	FLOPs (M)	latency (ms)	Params (M)	training (Gdays)	search (Gdays)
SCARLET-C	75.6	280	67	6.0	10	12
MnasNet-A1	75.2	312	55	3.9	288‡	-
GreedyNAS-C	**76.2**	284	70	4.7	7	< 1
FairNAS-C	74.7	321	75	4.4	10	2
SCARLET-B	76.3	329	104	6.5	10	12
GreedyNAS-B	**76.8**	324	110	5.2	7	< 1
SCARLET-A	76.9	365	118	6.7	10	12
EfficientNet-B0	76.3	390	82	5.3	-	-
DARTS	73.3	574	-	4.7	4†	-
GreedyNAS-A	**77.1**	366	77	6.5	7	< 1

Rank correlation coefficient of 1000 paths measured by the loss (ACC) of 1K vs 50K validation images w.r.t. different types of supernets.

Spearman rho			Kendall tau		
random	uniform(ACC)	greedy	random	uniform(ACC)	greedy
0.155	0.968(0.869)	**0.997**	0.113	0.851(0.699)	**0.961**