Stronger NAS with Weaker Predictors

- 作者：Junru Wu, Xiyang Dai, Dongdong Chen et.al.
- 机构：Texas A&M University, 微软, University of Texas at Austin
- 会议：NIPS2021
- 地址：https://arxiv.org/abs/2102.10490
- 代码：https://github.com/VITA-Group/WeakNAS

论文主要内容

摘要

标准的predict-based NAS通过有限样本数据拟合**整个搜索空间**，但很难定位top模型。本文提出逐渐采样更好的架构，逐步拟合一系列的predictor来关注top模型。

研究内容

Motivation

- Predicted-based NAS
  - 两个step：
    - 采样step：采样一些（architecture，performance）pairs
    - Fit step：用采样的数据来拟合得到一个预测器，预测整个空间的performance distribution
  - 已有的方法会构建一个**strong** predictor，对整个空间预测performance
  - 困难：
    - 由于架构空间的高维、非凸，（就算是strong predictor）很难用**有限的**样本拟合整个空间
    - 不同的predictor往往需要不同的架构representations/encode 来提升效果
- Predicted-based NAS：“如果最终目标是为了找最好的**那个**架构，我们需要对整个空间都建模的很好吗？”——用**weak** predictor拟合local space，然后**逐渐**把搜索空间朝着子空间(有good arch的位置)挪动
- 直觉上：整个空间可以划分成一些子空间，一部分相对好，一部分相对差。会选择好的那个子空间、丢掉差的子空间，让**更多样本**来自好空间。因为更需要good subspace上的predictor能力来鉴别最好的architecture

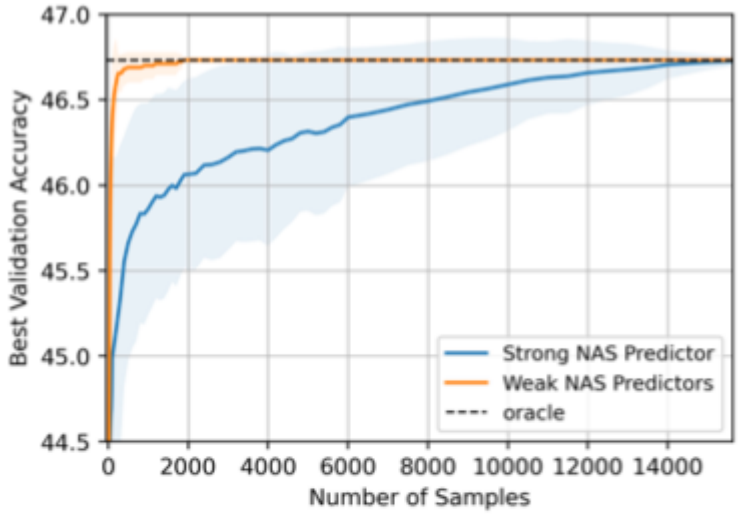


Figure 1: Comparison between our method using a set of weak predictors (iterative sampling), and a single strong predictor (random sampling) on NAS-Bench-201. For fair comparison, the NAS predictor in both methods adtops the same type of MLP described in 2.4. Solid lines and shadows denote the mean and standard deviation (std), respectively.

> ❗ 文章用的strong/weak predictor并不是用predictor的参数量来分，而是用采样方式。
>
> strong predictor：整个空间上均匀采样，拟合一个强predictor
>
> Weak predictor：一个predictor只负责预测一个subspace，与迭代采样策略关联

方法

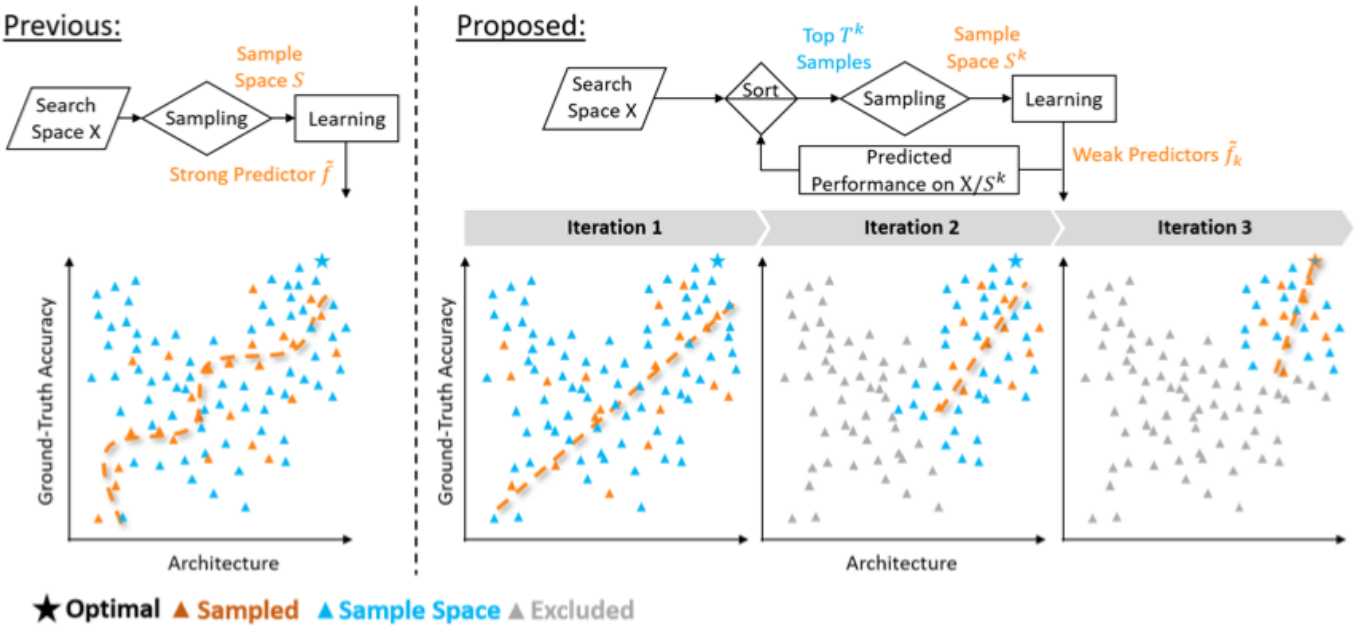


Figure 2: An illustration of WeakNAS's progressive approximation. Previous predictor-based NAS uniformly sample in the whole search space to fit a strong predictor. Instead, our method progressively shrinks the sample space based on predictions from previous weak predictors, and update new weak predictors towards subspace of better architectures, hence focusing on fitting the search path.

介绍

- 传统predictor-based：

$$x^* = \arg\max_{x \in X} \tilde{f}(x \mid S), \text{ s.t. } \tilde{f} = \arg\min_{S, \tilde{f} \in \tilde{\mathcal{F}}} \sum_{s \in S} \mathcal{L}(\tilde{f}(s), f(s))$$

1. 采样 $(s, f(s))$ pairs
2. 用loss $\mathcal{L}$ 来fitting predictor

Progressive Weak Predictors

$$\text{(Sampling)} \quad \tilde{P}^k = \{\tilde{f}_k(s) | s \in X \setminus S^k\},\ S_M \subset \text{Top}_N(\tilde{P}^k),\ S^{k+1} = S_M \cup S^k, \quad \text{where } \text{Top}_N(\tilde{P}^k) \text{ denote the set of top N architectures in } \tilde{P}^k \tag{3}$$

$$\text{(Predictor Fitting)} \quad x^* = \arg\max_{x \in X} \tilde{f}(x | S^{k+1}),\ \text{s.t. } \tilde{f}_{k+1} = \arg\min_{\tilde{f}_k \in \tilde{\mathcal{F}}} \sum_{s \in S^{k+1}} \mathcal{L}(\tilde{f}(s), f(s)) \tag{4}$$

Subproblem：架构采样

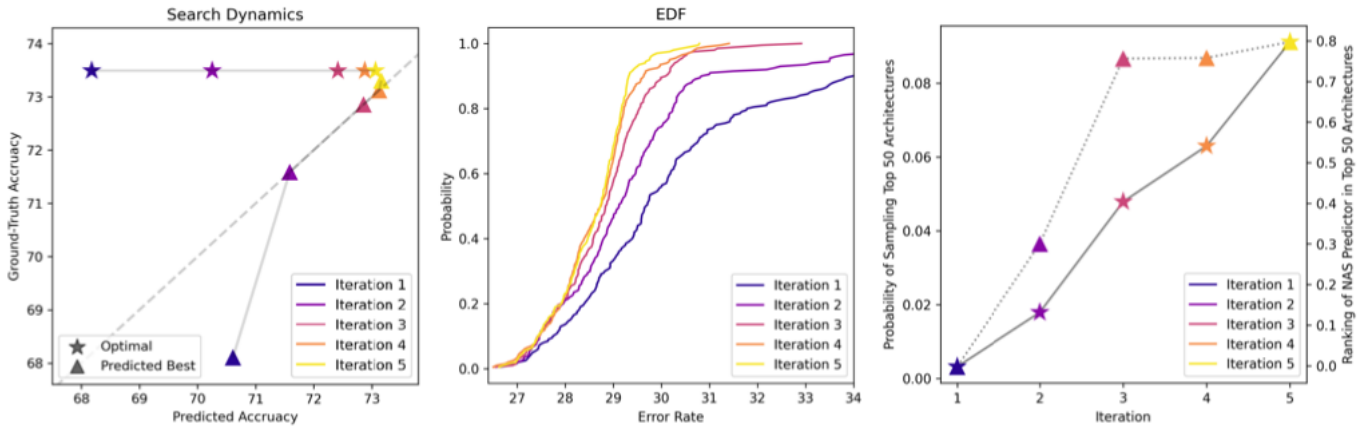
- 在iteration k+1，将整个空间 X (排除采样过的 S^k)用预测结果 $\tilde{P}^k$ 来排序。从TOP N 里面随机抽出 M 个新架构，query 它们的performance。

Subproblem：Weak predictor 拟合

- 拟合predictor $\tilde{f}^{k+1}$ ，并拿来得到预测结果 $\tilde{P}^{k+1}$

最终架构：最后一次迭代preditor的预测最好的那个架构

验证直觉（nasbench-201）



(a) (b) (c)

Figure 3: Visualization of the search dynamics in NAS-Bench-201 Search Space. (best viewed in color) (a) The trajectory of the predicted best architecture and global optimal through out 5 iterations; (b) Error *empirical distribution function* (EDF) of the predicted top-200 architectures throughout 5 iterations (c) Triangle marker: probability of sampling top-50 architectures throughout 5 iterations; Star marker: Kendall's Tau ranking of NAS predictor in Top 50 architectures through out 5 iterations.

- a) 图：optimal和predicted-optimal随着iteration靠的越来越近：preditor对最优架构的性能预测越来越准
- b) 图：predictor对Top-200架构的拟合效果越来越好（error显然会减小）
- c) 图：三角形说明每次迭代采样到gt-top50架构的概率变高；**五角星说明gt-TOP50架构的kdt提高：对top架构的甄别能力提升**

实际每次迭代只用一个预测器，最后结果也用一个预测器，有没有可能因为样本不平衡，导致最后一predictor把一个差架构预测到top1（对差架构拟合效果不好）/或者说如何**保证**差架构不会被最后选出

Why it works

可以看作bayes optimization

$acq(x) = u(x-\theta)\cdot\epsilon$ ，其中 ϵ 服从 $\dot{U}(0,1)$ ， $u(x)$ =1 若x ⩾ θ，否则0

- θ 可以看作是划分TOP N的阈值

采样： $S_M = \underset{\text{Top } M}{\arg\max} acq(x)$

由此得到： "*oversimplified BO*"

NAS空间内在的结构化特点

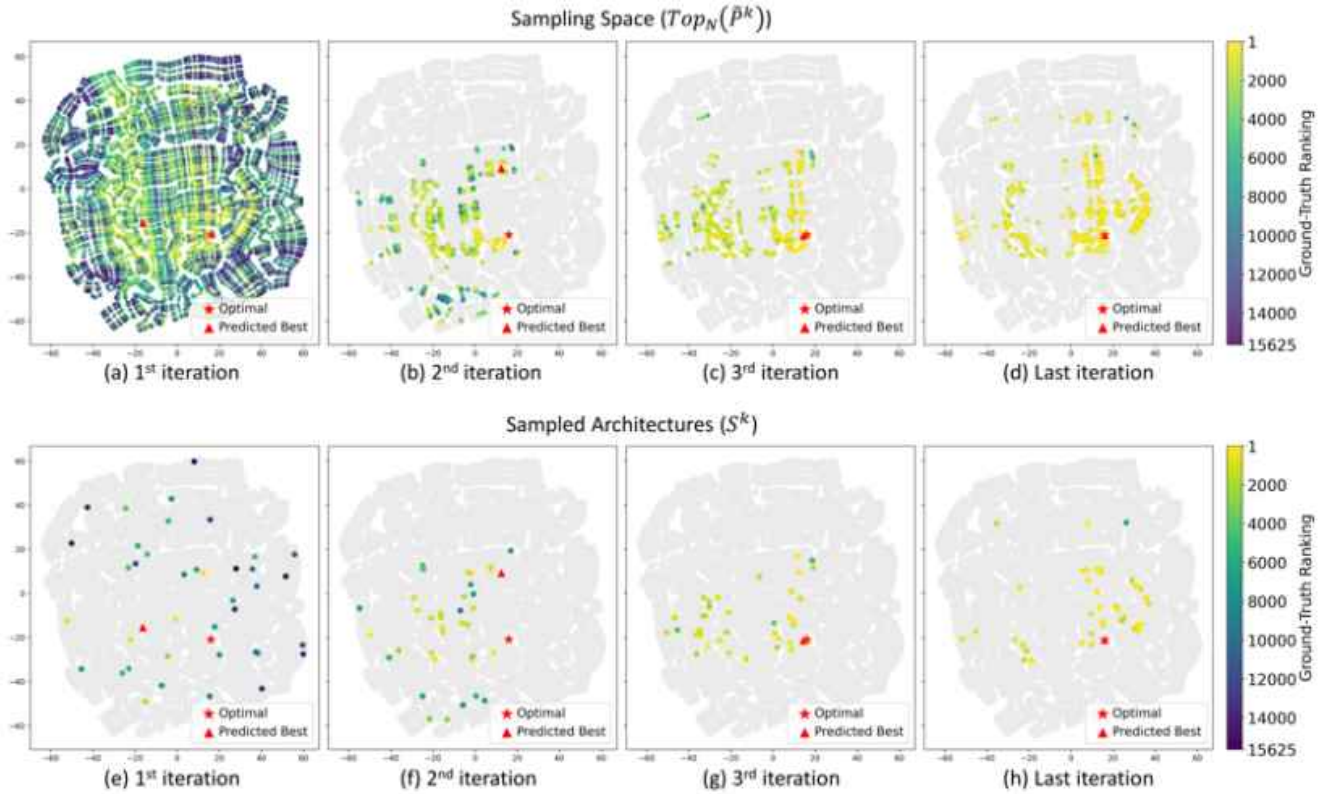


Figure 4: Visualization of search dynamics in NAS-Bench-201 Search Space via t-SNE. At i-th iteration, we randomly sample M = 40 new architectures from the top N = 400 ranked architectures in $\tilde{P}^k$. The top row from (a)-(d) show the sampling space $Top_N(\tilde{P}^k)$, and the bottom row from (e)-(h) show the sampled architectures S^k. The performance ranking of architectures is encoded by color, and those not-sampled architectures are colored in grey.

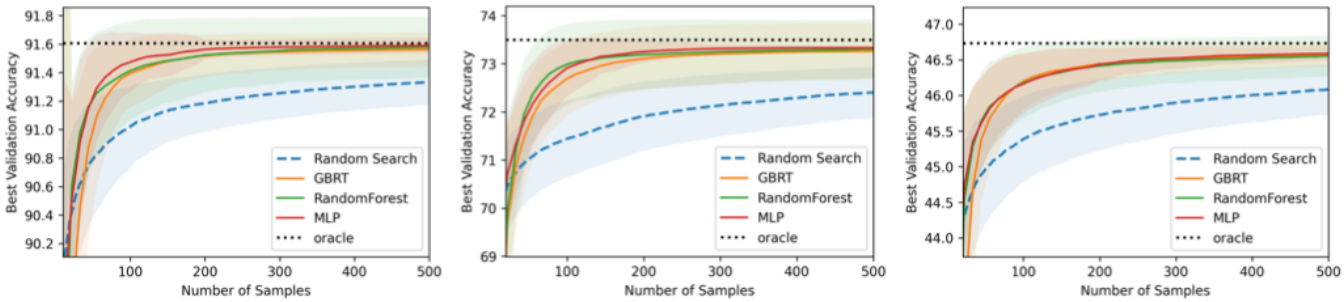
- highly-structured、best arch 和best arch互相靠在一起

exploration-exploitation trade-off

- 开始的时候，weak predictor粗略的拟合整个空间，采样中noisier => exploration
- 后面阶段，weak predictor在表现好的簇内拟合的很好=> exploitation
- 每次采样是从预测的TOP N中随机抽M个架构，M/N这个比例自带trade-off；随机抽M个的策略也可以是uniform, linear-decay or exponential-decay

predictor选择

- predictor：
  - MLP
  - *Regression Tree：GBRT*
  - *Random Forest*
- feature，用来encode架构：
  - *One-hot vector*
  - *Adjacency matrix*



(a) CIFAR10 (b) CIFAR100 (c) ImageNet16-120

Figure 5: Evaluations of robustness across different predictors on NAS-Bench-201. Solid lines and shadow regions denote the mean and std, respectively.

对predictor、feature的选择很广泛、鲁棒

实验

NASbench-101、NASbench-201、

Open Domain Search：NASNet search space、MobileNet search space

- 由于open domain search用sample-based计算开销很大，使用权重共享的超网performance proxy来代替gt
- MobileNet 使用pretrained OFA
- NASNet使用Single-Path One-shot

Results

消融实验

1. 迭代采样的有效性：

Sampling	#Predictor	#Queries	Test Acc.(%)	SD(%)	Test Regret(%)	Avg. Rank
Uniform	1 Strong Predictor	2000	93.92	0.08	0.40	135.0
Iterative	1 Weak Predictor	100	93.42	0.37	0.90	6652.1
	11 Weak Predictors	200	94.18	0.14	0.14	5.6
	91 Weak Predictors	1000	94.25	0.04	0.07	1.7
	191 Weak Predictors	2000	94.26	0.04	0.06	1.6
Optimal	-	-	94.32	-	0.00	1

Table 1: Ablation on the effectiveness of our iterative scheme on NAS-Bench-101

2. N/M，探索开发 trade-off

Sampling (M from TopN)	M	TopN	#Queries	Test Acc.(%)	SD(%)	Test Regret(%)	Avg. Rank
Exponential-decay	10	100	1000	93.96	0.10	0.36	85.0
Linear-decay	10	100	1000	94.06	0.08	0.26	26.1
Uniform	**10**	**100**	**1000**	**94.25**	**0.04**	**0.07**	**1.7**
Uniform	10	1000	1000	94.10	0.19	0.22	14.1
Uniform	**10**	**100**	**1000**	**94.25**	**0.04**	**0.07**	**1.7**
Uniform	10	10	1000	94.24	0.04	0.08	1.9

Table 2: Ablation on exploitation-exploration trade-off on NAS-Bench-101

3. 使用经典EI的BO变体

Method	#Queries	Test Acc.(%)	SD(%)	Test Regret(%)	Avg. Rank
WeakNAS	**1000**	**94.25**	**0.04**	**0.07**	**1.7**
WeakNAS (BO Variant)	1000	94.12	0.15	0.20	8.7
Optimal	-	94.32	-	0.00	1.0

Table 3: Comparing to the BO variant of WeakNAS on NAS-Bench-101.

对比SOTA

Method	#Queries	Test Acc.(%)	SD(%)	Test Regret(%)	Avg. Rank
Random Search	2000	93.64	0.25	0.68	1750.0
NAO [2]	2000	93.90	0.03	0.42	168.1
Reg Evolution [14]	2000	93.96	0.05	0.36	85.0
Semi-NAS [20]	2000	94.02	0.05	0.30	42.1
Neural Predictor [7]	2000	94.04	0.05	0.28	33.5
WeakNAS	**2000**	**94.26**	**0.04**	**0.06**	**1.6**
Semi-Assessor [42]	1000	94.01	-	0.31	47.1
LaNAS [21]	1000	94.10	-	0.22	14.1
BONAS [19]	1000	94.22	-	0.10	3.0
WeakNAS	**1000**	**94.25**	**0.04**	**0.07**	**1.7**
Arch2vec [41]	400	94.10	-	0.22	14.1
WeakNAS	**400**	**94.24**	**0.04**	**0.08**	**1.9**
LaNAS [21]	200	93.90	-	0.42	168.1
BONAS [19]	200	94.09	-	0.23	18.0
WeakNAS	**200**	**94.18**	**0.14**	**0.14**	**5.6**
NASBOWLr [45]	150	94.09	-	0.23	18.0
CATE (cate-DNGO-LS) [43]	150	94.10	-	0.22	12.3
WeakNAS	**150**	**94.10**	**0.19**	**0.22**	**12.3**
Optimal	-	94.32	-	0.00	1.0

Table 4: Comparing searching efficiency by limiting the total query amounts on NAS-Bench-101.

Method	NAS-Bench-101	NAS-Bench-201		
Dataset	CIFAR10	CIFAR10	CIFAR100	ImageNet16-120
Random Search	188139.8	7782.1	7621.2	7726.1
Reg Evolution [14]	87402.7	563.2	438.2	715.1
MCTS [40]	73977.2	†528.3	†405.4	†578.2
Semi-NAS [20]	†47932.3	-	-	-
LaNAS [21]	11390.7	†247.1	†187.5	†292.4
BONAS [19]	1465.4	-	-	-
WeakNAS	**195.2**	**182.1**	**78.4**	**268.4**

Table 5: Comparison on the number of samples required to find the global optimal on NAS-Bench-101 and NAS-Bench-201. † denote reproduced results using adapted code.

Open domain

Model	Queries(#)	Top-1 Err.(%)	Top-5 Err.(%)	Params(M)	FLOPs(M)	GPU Days
MobileNetV2	-	25.3	-	6.9	585	-
ShuffletNetV2	-	25.1	-	5.0	591	-
SNAS[51]	-	27.3	9.2	4.3	522	1.5
DARTS[1]	-	26.9	9.0	4.9	595	4.0
P-DARTS[52]	-	24.4	7.4	4.9	557	0.3
PC-DARTS[53]	-	24.2	7.3	5.3	597	3.8
DS-NAS[53]	-	24.2	7.3	5.3	597	10.4
NASNet-A [35]	20000	26.0	8.4	5.3	564	2000
AmoebaNet-A [14]	10000	25.5	8.0	5.1	555	3150
PNAS [46]	1160	25.8	8.1	5.1	588	200
NAO [2]	1000	24.5	7.8	6.5	590	200
LaNAS [21] (Oneshot)	800	24.1	-	5.4	567	3
LaNAS [21]	800	23.5	-	5.1	570	150
WeakNAS	**800**	**23.5**	**6.8**	**5.5**	**591**	**2.5**

Table 6: Comparison to SOTA results on ImageNet using NASNet search space.

Model	Queries(#)	Top-1 Acc.(%)	Top-5 Acc.(%)	FLOPs(M)	GPU Days*
Proxyless NAS[54]	-	75.1	92.9	-	-
Semi-NAS[20]	300	76.5	93.2	599	-
BigNAS[47]	-	76.5	-	586	-
FBNetv3[48]	20000	80.5	95.1	557	-
OFA[36]	16000	80.0	-	595	1.6
LaNAS[21]	800	80.8	-	598	0.3
WeakNAS	**1000**	**81.3**	**95.1**	**560**	**0.16**
	800	**81.2**	**95.2**	**593**	**0.13**

Table 7: Comparison to SOTA results on ImageNet using MobileNet search space. *Does not include supernet training cost.