Neural Architecture Search

REDS

Charles VIN, Mathis KOROGLU

Sorbonne Université

February 7, 2024

Overview

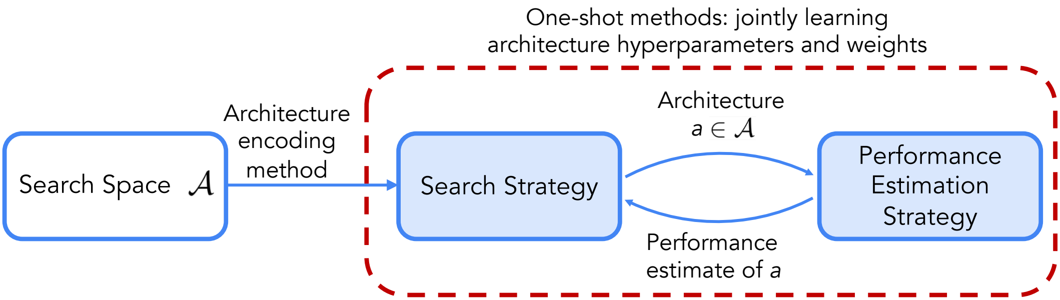


Figure: Overview of NAS.
A search strategy iteratively selects architectures (typically by using an architecture encoding method) from a predened search space $\mathcal{A}$.
The architectures are passed to a performance estimation strategy, which returns the performance estimate to the search strategy.
For one-shot methods, the search strategy and performance estimation strategy are inherently coupled.

Search space

Definition

The set of all architectures that the NAS algorithm is allowed to select.

- Size: from a few thousand to over 10^{20}.
- Reduction: adding domain knowledge.

$\rightarrow$ Introduce humain bias $\rightarrow x$ reduce the chance of finding truly nover architecture.

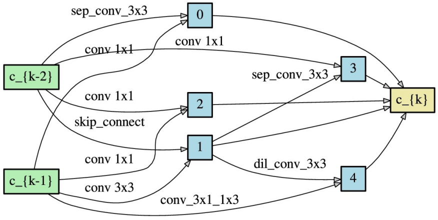


Figure: Architecture directed acyclic graph (DAG) with operation on nodes

Search strategy

Definition

The optimization technique used to find a high-performing architecture in the search space.

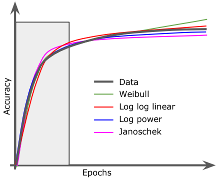
- Black-box optimization techniques : RL, Bayesian optimization, evolutionary search.
- One-shot techniques: supernet-hypernet based methods.

Performance estimation strategy

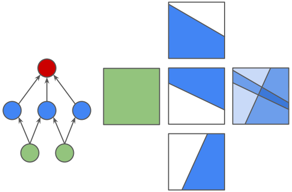
Definition

Any method used to quickly predict the performance of neural architectures in order to avoid fully training the architecture.

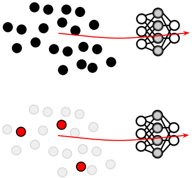
- Full training & evaluation.
- Performance estimation strategy.



Learning Curve Extrapolation



Zero-Cost Proxies



Subset Selection

Benchmarks

Definition

A NAS benchmark is defined as a dataset with a fixed train-test split, a search space, and a fixed evaluation pipeline for training the architectures.

- Tabular benchmarks: precomputed evaluations for all possible architecture in the search space.

→ Allow to **simulate** hundreds of trials !
  - No more training.
  - Statistically significant comparisons (simulation with multiple seeds).
- Specialization: precomputed evaluation for NLP tasks, protein folding, astronomy imaging, for vision dataset (CIFAR-10(0), ImageNet, ...)
- Most popular NAS benchmarks
  - NAS-Bench-101/202
  - Surr-NAS-Bench-DARTS
  - TransNAS-Bench-101
  - NAS-Bench-Suite

Evalutation

Table 2: **Comparison with Transferable NAS on NB201 Serach Space.** We present the accuracy achieved on four unseen datasets. Additionally, we provide the number of neural architectures (**Trained Archs**) that are actually trained to achieve accuracy. The accuracies are reported with 95% confidence intervals over 3 runs.

Type	Method	CIFAR-10 Accuracy (%)	CIFAR-10 Trained Archs	CIFAR-100 Accuracy (%)	CIFAR-100 Trained Archs	Aircraft Accuracy (%)	Aircraft Trained Archs	Oxford-IIIT Pets Accuracy (%)	Oxford-IIIT Pets Trained Archs
	ResNet (He et al., 2016)	$93.97_{\pm 0.00}$	N/A	$70.86_{\pm 0.00}$	N/A	$47.01_{\pm 1.16}$	N/A	$25.58_{\pm 3.43}$	N/A
	RS (Bergstra & Bengio, 2012)	$93.70_{\pm 0.36}$	> 500	$71.04_{\pm 1.07}$	> 500	-	-	-	-
	REA (Real et al., 2019)	$93.92_{\pm 0.30}$	> 500	$71.84_{\pm 0.99}$	> 500	-	-	-	-
	REINFORCE (Williams, 1992)	$93.85_{\pm 0.37}$	> 500	$71.71_{\pm 1.09}$	> 500	-	-	-	-
One-shot NAS*	RSPS (Li & Talwalkar, 2019)	$84.07_{\pm 3.61}$	N/A	$52.31_{\pm 5.77}$	N/A	$42.19_{\pm 3.88}$	N/A	$22.91_{\pm 1.65}$	N/A
	SETN (Dong & Yang, 2019a)	$87.64_{\pm 0.00}$	N/A	$59.09_{\pm 0.24}$	N/A	$44.84_{\pm 3.96}$	N/A	$25.17_{\pm 1.68}$	N/A
	GDAS (Dong & Yang, 2019b)	$93.61_{\pm 0.09}$	N/A	$70.70_{\pm 0.30}$	N/A	$53.52_{\pm 0.48}$	N/A	$24.02_{\pm 2.75}$	N/A
	PC-DARTS (Xu et al., 2020)	$93.66_{\pm 0.17}$	N/A	$66.64_{\pm 2.34}$	N/A	$26.33_{\pm 3.40}$	N/A	$25.31_{\pm 1.38}$	N/A
	DrNAS (Chen et al., 2021)	$94.36_{\pm 0.00}$	N/A	$\mathbf{73.51_{\pm 0.00}}$	N/A	$46.08_{\pm 7.00}$	N/A	$26.73_{\pm 2.61}$	N/A
BO-based NAS	BOHB (Falkner et al., 2018)	$93.61_{\pm 0.52}$	> 500	$70.85_{\pm 1.28}$	> 500	-	-	-	-
	GP-UCB	$\mathbf{94.37_{\pm 0.00}}$	58	$73.14_{\pm 0.00}$	100	$41.72_{\pm 0.00}$	40	$40.60_{\pm 1.10}$	11
	BANANAS (White et al., 2021a)	$\mathbf{94.37_{\pm 0.00}}$	46	$\mathbf{73.51_{\pm 0.00}}$	88	$41.72_{\pm 0.00}$	40	$40.15_{\pm 1.59}$	17
	NASBOWL (Ru et al., 2021)	$94.34_{\pm 0.00}$	100	$\mathbf{73.51_{\pm 0.00}}$	87	$53.73_{\pm 0.83}$	40	$41.29_{\pm 1.10}$	17
	HEBO (Cowen-Rivers et al., 2022)	$94.34_{\pm 0.00}$	100	$72.62_{\pm 0.20}$	100	$49.32_{\pm 6.10}$	40	$40.55_{\pm 1.15}$	18
Transferable NAS	TNAS (Shala et al., 2023)	$\mathbf{94.37_{\pm 0.00}}$	29	$\mathbf{73.51_{\pm 0.00}}$	59	$59.15_{\pm 0.58}$	26	$40.00_{\pm 0.00}$	6
	MetaD2A (Lee et al., 2021a)	$\mathbf{94.37_{\pm 0.00}}$	100	$73.34_{\pm 0.04}$	100	$57.71_{\pm 0.20}$	40	$39.04_{\pm 0.20}$	40
	DiffusionNAG (Ours)	$\mathbf{94.37_{\pm 0.00}}$	**5**	$\mathbf{73.51_{\pm 0.00}}$	**5**	$\mathbf{59.63_{\pm 0.92}}$	**2**	$\mathbf{41.32_{\pm 0.84}}$	**2**

Figure: Result table example

Evalutation

Table 8
Comparison of the evaluation results on CIFAR-10 and CIFAR-100.

Architecture	Test Err. (%) C-10	Test Err. (%) C-100	Params (.M)	Search Cost (GPU-days)	Search Method
NASNet-A [8]	2.65	-	3.3	1800	RL
AmoebaNet-A [20]	3.34	-	3.2	3150	Evolution
PNAS [40]	3.41	-	3.2	225	SMBO
RelativeNAS [41]	2.34	15.86	3.93	0.4	Evolution
DARTS (first order) [10]	3	17.76	3.3	1.5	Gradient
SNAS + mild constraint [42]	2.98	-	2.9	1.5	Gradient
ProxylessNAS [29]	2.08	-	5.7	4	Gradient
P-DARTS† (C-10) [18]	2.5	16.55	3.4	0.3	Gradient
P-DARTS† (C-100) [18]	2.62	15.92	3.6	0.3	Gradient
P-DARTS† (C-10-Large) [18]	2.25	15.27	10.5	0.3	Gradient
P-DARTS† (C-100-Large) [18]	2.43	14.64	11	0.3	Gradient
Ours† (`C-10`)	2.47 ± 0.03	-	2.04	$1.3^{\star}$	RL
Ours† (`C-100`)	2.58 ± 0.05	-	3.43	$1.3^{\star}$	RL
Ours† (`C-10-Large`)	-	15.3 ± 0.04	9.57	$1.3^{\star}$	RL
Ours† (`C-100-Large`)	-	14.6 ± 0.03	10.5	$1.3^{\star}$	RL

Figure: Result table example

Contribution

- Problem: Classical NAS need to train thousand of architecture.

$\rightarrow$ Switch from searching architecture to directly generate architecture using a graph diffusion model.

$\rightarrow$ diffusionNAG

Contribution

Contribution

Contribution

Limits

- Need to retrain f_Φ for every metrics
  - Accuracy
  - Inference time
  - Memory usage
  - Adversarial attack resistance
  - ...
- Only one metric is optimized: no compromise possible
- Linear combination of metrics: not ideal to find a good compromise

Our contrib title

- Train a multi-objective predictor using the same task aware dataset, but enhanced with metrics of interest
- Guide the diffusion with the multi-objective predictor:
  - Generate a bigger super-network with disableable blocks
  - ie: equal to an encoding of points near the pareto front (best possible compromise between metrics)
- Constraints encoding within the same score network
  - Layer compatibility in a disableable block
  - Compatibility between blocks inputs and outputs

How to use it :

- Train the obtained super-network after diffusion inference
- disable the blocks depending on the metrics of interest

Advantages and limits of this approach

- Flexibility for the user : fix the metrics threshold only after performing the NAS
- Dynamic architecture depending on external parameters
- But predictor needs to be retrained if a new metric is added, and diffusion inference more costly
- No guarantees if the pareto optimality is well covered by the disableable blocks

Multiple Columns

Heading

1. Statement
2. Explanation
3. Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lectus nisl, ultricies in feugiat rutrum, porttitor sit amet augue. Aliquam ut tortor mauris. Sed volutpat ante purus, quis accumsan dolor.

References

John Smith (2012)
Title of the publication
Journal Name 12(3), 45 – 678.

Thank you