PASHA: Efficient HPO and NAS with Progressive Resource Allocation

Ondrej Bohdal[1*], Lukas Balles[2], Martin Wistuba[2], Beyza Ermis[3*], Cedric Archambeau[2], Giovanni Zappella[2]

THE UNIVERSITY of EDINBURGH informatics [1]

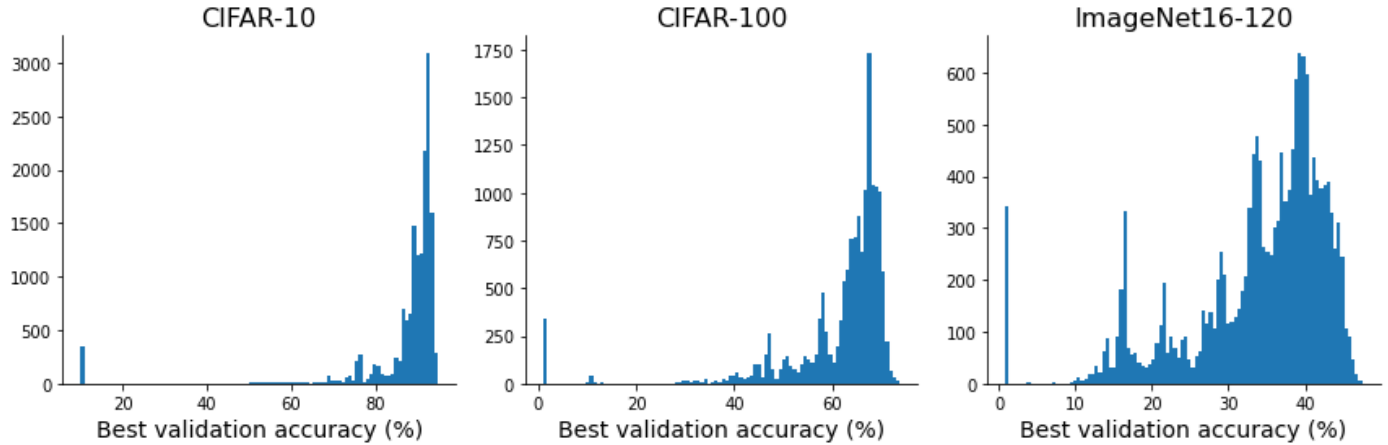
amazon | science [2]

Cohere For AI [3]

* work done while at AWS, Berlin

HPO and NAS

- Selection of hyperparameters and neural network architecture has a large impact on performance of the model
- Example: various architectures in NASBench201 have significantly different performances



Challenges with Large-Scale HPO and NAS

- Costs make large-scale HPO difficult and often unviable
  - Evaluating 50 configurations for a 340-million-parameter BERT model (Devlin et al., NAACL'19) on the 15GB Wikipedia and Book corpora would cost around $500,000
- Multi-fidelity methods such as ASHA (Li et al., MLSys'20) require specifying max amount of resources
- How to specify the max amount of resources?
  - Usually overestimated to guarantee convergence
  - Excellent configurations could be found using far fewer resources
- Insight: ranking of configurations is relatively stable after initial part of training
  - Learning curves rarely cross in later stages of training (excluding noise)

PASHA

- Variation on ASHA (Li et al., MLSys'20) - asynchronous successive halving
- Idea: dynamically increase max resources depending on if ranking of configurations is stable
  - Start with a small initial amount of maximum resources
  - Progressively increase them if the ranking of the configurations in the top two rungs (rounds of promotion) has not stabilized
  - Due to stochasticity, some benevolence in rankings needed → use soft ranking
- Particularly useful for HPO on massive datasets

Illustration

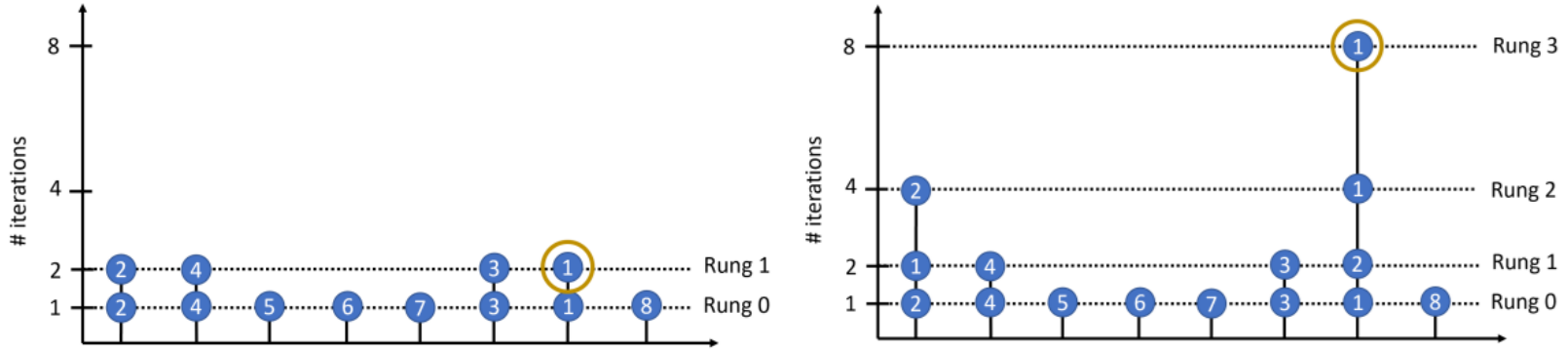


Figure 1: Illustration of how PASHA stops early if the ranking of configurations has stabilized. Left: the ranking of the configurations (displayed inside the circles) has stabilized, so we can select the best configuration and stop the search. Right: the ranking has not stabilized, so we continue.

Soft Ranking

- Configurations are equivalent if their performance difference is less than ϵ
  - Estimate ϵ based on noise in rankings across epochs
  - Intuition: configurations that repeatedly swap their rankings are similar

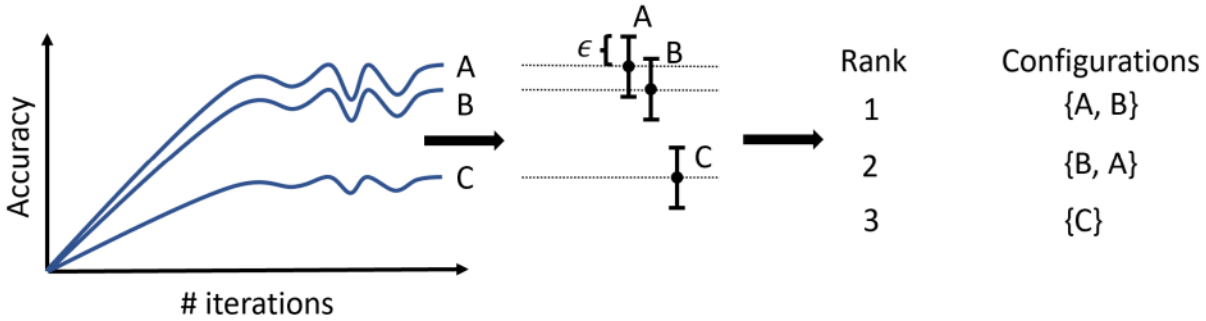


Figure 2: Illustration of soft ranking. There are three lists with the first two containing two items because the scores of the two configurations are closer to each other than ϵ.

Results - NAS

Table 1: NASBench201 results. PASHA leads to large improvements in runtime, while achieving similar accuracy as ASHA.

Dataset	Approach	Accuracy (%)	Runtime	Speedup factor	Max resources
CIFAR-10	ASHA	93.85 ± 0.25	3.0h ± 0.6h	1.0x	200.0 ± 0.0
	PASHA	93.57 ± 0.75	1.3h ± 0.6h	2.3x	36.1 ± 50.0
	One-epoch baseline	93.30 ± 0.61	0.3h ± 0.0h	8.5x	1.0 ± 0.0
	Random baseline	72.88 ± 19.20	0.0h ± 0.0h	N/A	0.0 ± 0.0
CIFAR-100	ASHA	71.69 ± 1.05	3.2h ± 0.9h	1.0x	200.0 ± 0.0
	PASHA	71.84 ± 1.41	0.9h ± 0.4h	3.4x	20.5 ± 48.3
	One-epoch baseline	65.57 ± 5.53	0.3h ± 0.0h	9.2x	1.0 ± 0.0
	Random baseline	42.83 ± 18.20	0.0h ± 0.0h	N/A	0.0 ± 0.0
ImageNet16-120	ASHA	45.63 ± 0.81	8.8h ± 2.2h	1.0x	200.0 ± 0.0
	PASHA	45.13 ± 1.51	2.9h ± 1.7h	3.1x	21.3 ± 48.1
	One-epoch baseline	41.42 ± 4.98	1.0h ± 0.0h	8.8x	1.0 ± 0.0
	Random baseline	20.75 ± 9.97	0.0h ± 0.0h	N/A	0.0 ± 0.0

Combination with Bayesian Optimization

Table 2: NASBench201 results for ASHA with Bayesian Optimization searcher – MOBSTER (Klein et al., 2020) and similarly extended version of PASHA. The results show PASHA can be successfully combined with a smarter configuration selection strategy.

Dataset	Approach	Accuracy (%)	Runtime	Speedup factor	Max resources
CIFAR-10	MOBSTER	94.21 ± 0.28	$5.0\text{h} \pm 1.1\text{h}$	1.0x	200.0 ± 0.0
	PASHA BO	94.00 ± 0.20	$2.6\text{h} \pm 1.8\text{h}$	2.0x	70.7 ± 81.6
CIFAR-100	MOBSTER	72.79 ± 0.68	$5.7\text{h} \pm 1.4\text{h}$	1.0x	200.0 ± 0.0
	PASHA BO	72.16 ± 1.07	$1.6\text{h} \pm 0.5\text{h}$	3.7x	13.0 ± 8.7
ImageNet16-120	MOBSTER	46.21 ± 0.70	$15.1\text{h} \pm 4.0\text{h}$	1.0x	200.0 ± 0.0
	PASHA BO	45.36 ± 1.06	$3.9\text{h} \pm 1.2\text{h}$	3.9x	11.8 ± 7.9

Results - HPO on Large Datasets

Table 3: Results of the HPO experiments on WMT and ImageNet tasks from the PD1 benchmark. Mean and std of the best validation accuracy (or its equivalent as given in the PD1 benchmark).

Dataset	Approach	Accuracy (%)	Runtime	Speedup factor	Max resources
WMT	ASHA	62.72 ± 1.41	43.7h ± 37.2h	1.0x	1357.4 ± 80.4
	PASHA	62.04 ± 2.05	2.8h ± 0.6h	15.5x	37.8 ± 21.6
	One-epoch baseline	62.36 ± 1.40	0.6h ± 0.0h	67.3x	1.0 ± 0.0
	Random baseline	33.93 ± 21.96	0.0h ± 0.0h	N/A	0.0 ± 0.0
ImageNet	ASHA	75.10 ± 2.03	7.3h ± 1.2h	1.0x	251.0 ± 0.0
	PASHA	73.37 ± 2.71	3.8h ± 1.0h	1.9x	45.0 ± 30.1
	One-epoch baseline	63.40 ± 9.91	1.1h ± 0.0h	6.7x	1.0 ± 0.0
	Random baseline	36.94 ± 31.05	0.0h ± 0.0h	N/A	0.0 ± 0.0

Summary

- PASHA dynamically selects the amount of maximum resources
- Significant speedup of HPO and NAS without sacrificing the performance
  - Especially on large datasets
- Can be combined with Bayesian Optimization search strategies
- PASHA is available within Syne Tune HPO framework: https://github.com/awslabs/syne-tune
- Tutorial for PASHA is also available: https://syne-tune.readthedocs.io/en/latest/tutorials/pasha/pasha.html