In European Conference on Computer Vision (ECCV), 2020

Neural Predictor for Neural Architecture Search

Wei Wen[1,2], Hanxiao Liu[1], Yiran Chen[2], Hai Li[2], Gabriel Bender[1], Pieter-Jan Kindermans[1]

[1]*Google Brain*, [2]*Duke University*

20205642_**Patara Trirat**
20190754_**Guntitat Sawadwuthikul**

TEAM 3_**TH^2** / 2021.06.15

Contents

Contents

Baseline Paper's

Executive Summary

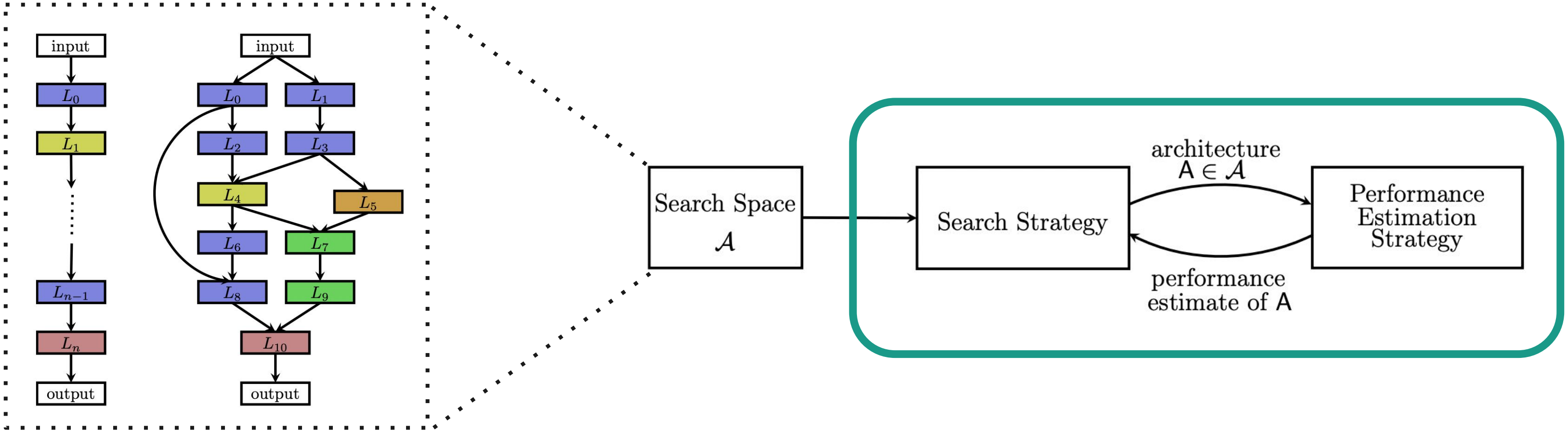
The baseline paper proposes, ***Neural Predictor***, an alternative solution for reducing the computation cost of neural architecture search. The neural predictor filters only a few high-potential architectures which will be validated manually to find the best one. In summary, the authors:

- Construct a neural predictor to predict the quality of architectures using GCNs
- Validate its performance on NAS-Bench-101 and ProxylessNAS search spaces
- Compare the results with other baselines, including the state-of-the-art
- Conduct an ablation study and propose frontier models for mobile devices

Contents

Introduction

Motivation

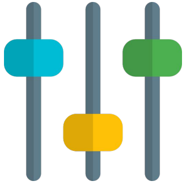


Given a neural network ***search space***, the task is to find the best architecture according to its validation performance.

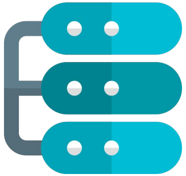
Introduction

Research Objectives

Sample ***Efficiency***



Simplicity *of Hyperparameter Tuning*

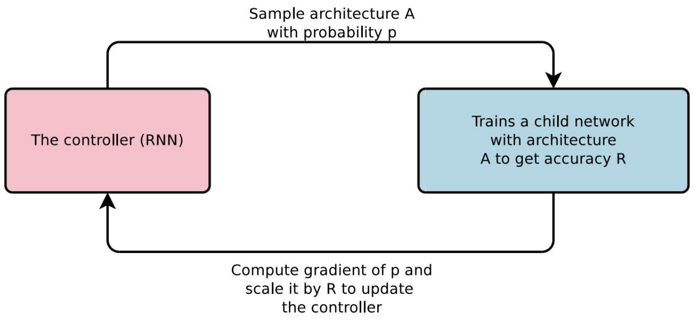


Full ***Parallelizability***

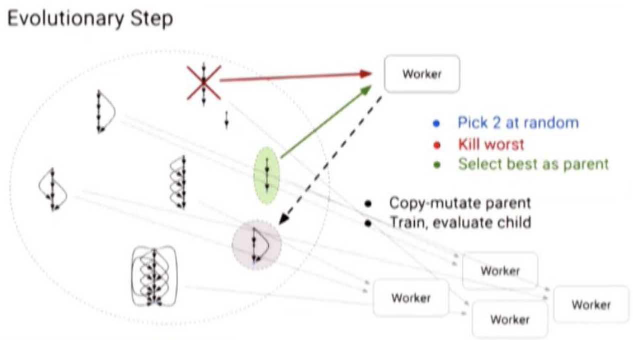
Contents

Related Work

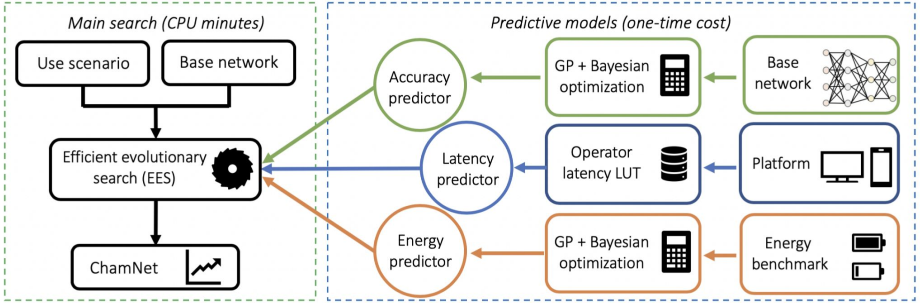
Sampling-based NAS



RL [1]



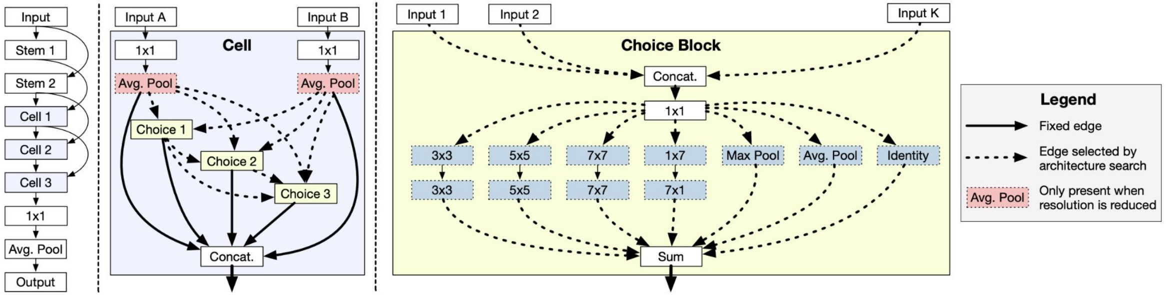
EA [2]

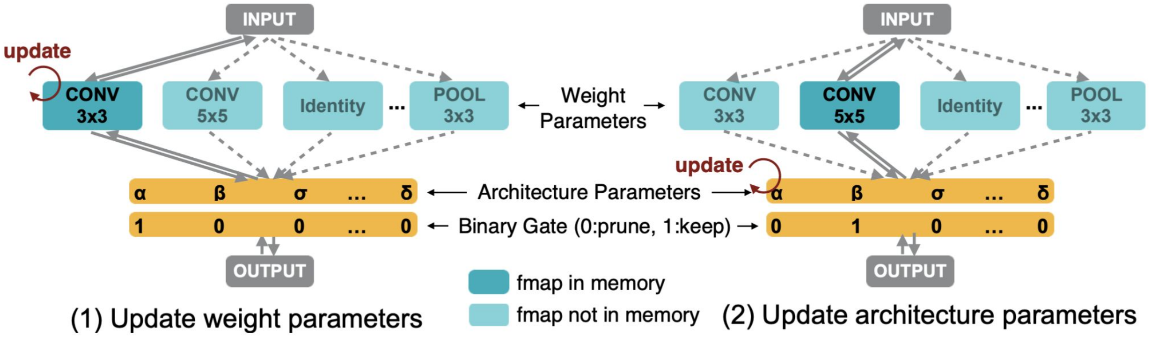


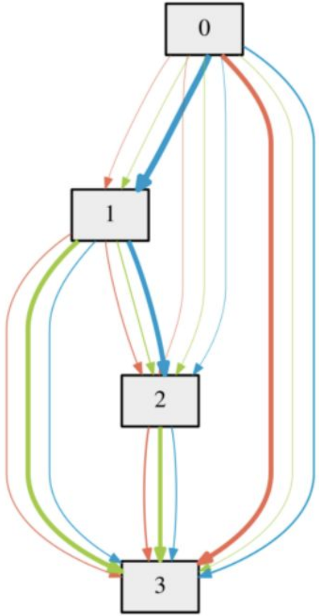
BO [3, 4]

Related Work

Weight/Parameter Sharing-based NAS





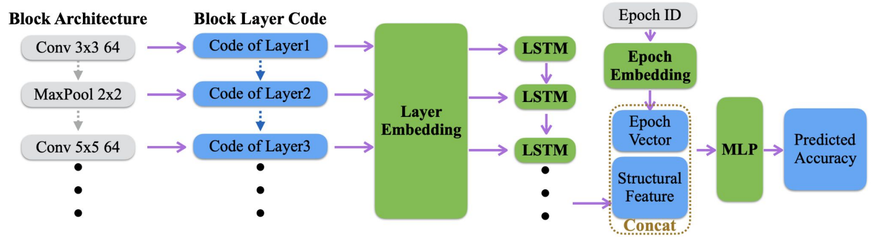


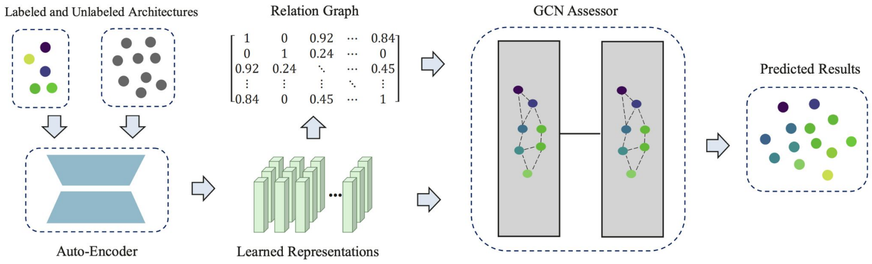
Weight Sharing (One-shot, DARTS)

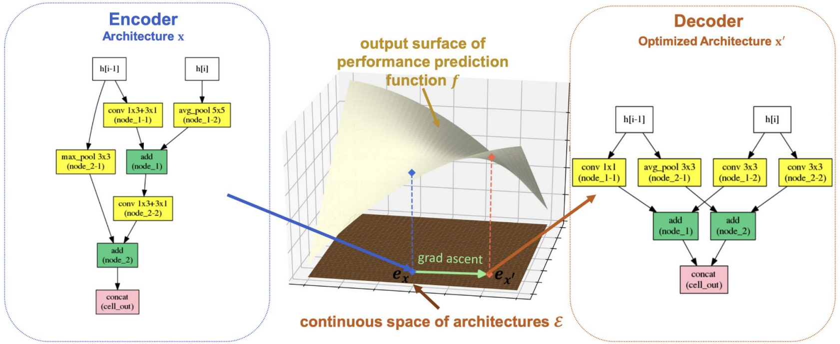
[5-9]

Related Work

Prediction-enhanced NAS







Performance Prediction [10-15]

Related Work

Shortcomings

(1) **training** thousands of models **from scratch**,
(2) **tuning hyperparameters** for every model, and
(3) **parallelizing** traditional algorithms (e.g., RL or EA), *are inevitably **expensive**.*

Methods	Efficient	Hyperparameter Friendly	Fully Parallelizable
Sampling-based (RL, EA, BO)	✘	✘	✘
Weight sharing (DARTS, One-shot)	✔	✘	✘
Neural Predictor	✔	✔	✔

Contents

Solution

Methodology Overview: Neural Predictor

Supervised Learning + Random Sampling

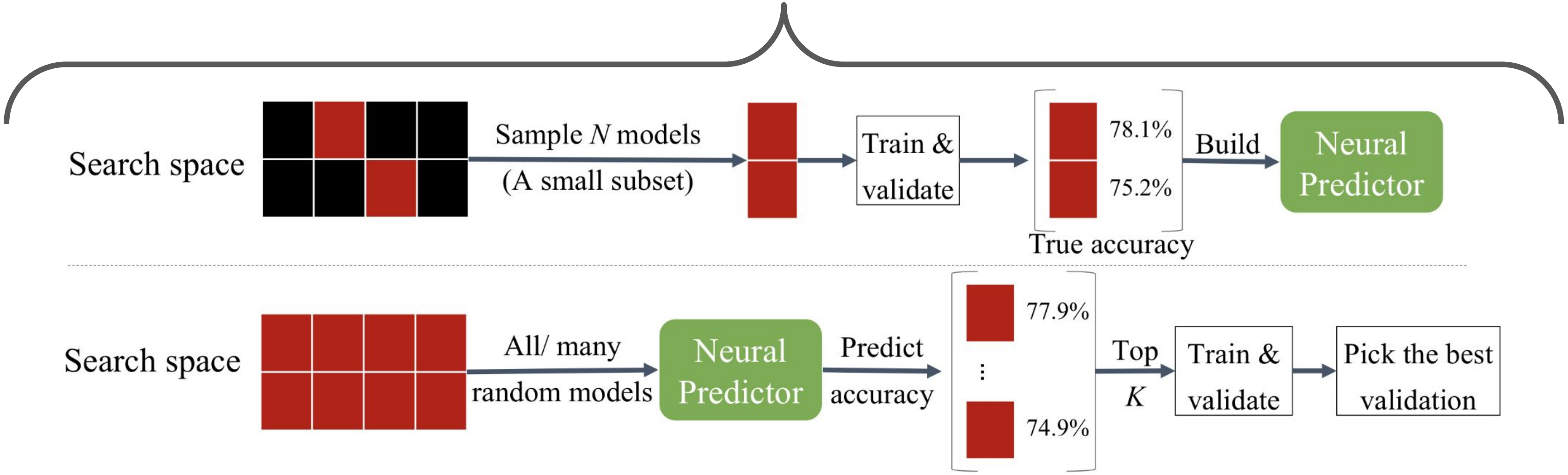
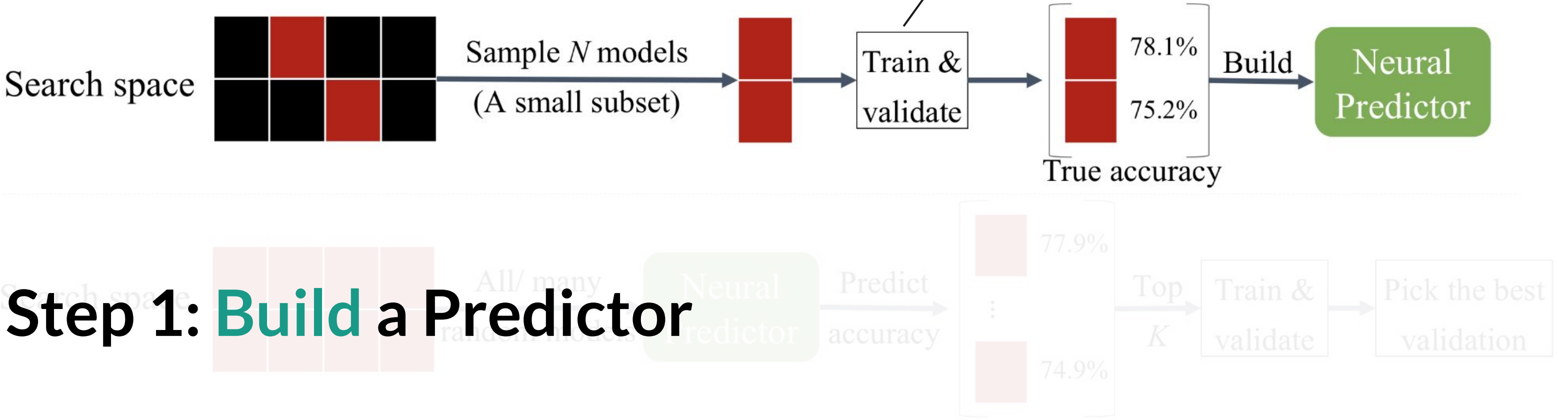


Fig. 1: Building (top) and applying (bottom) the Neural Predictor.

Solution

Neural Predictor

(architecture, validation accuracy) pairs



Step 1: Build a Predictor

Fig. 1: Building (top) and applying (bottom) the Neural Predictor.

Solution

Neural Predictor

Step 1: Build a Predictor

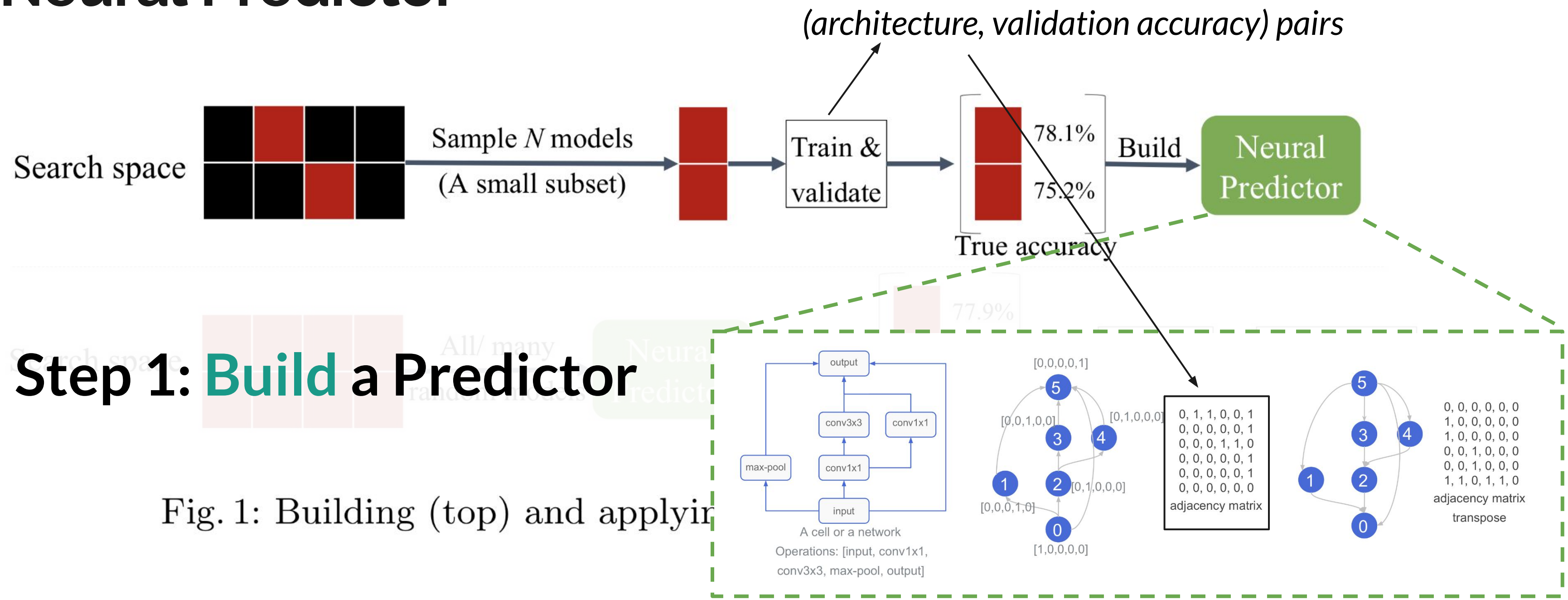


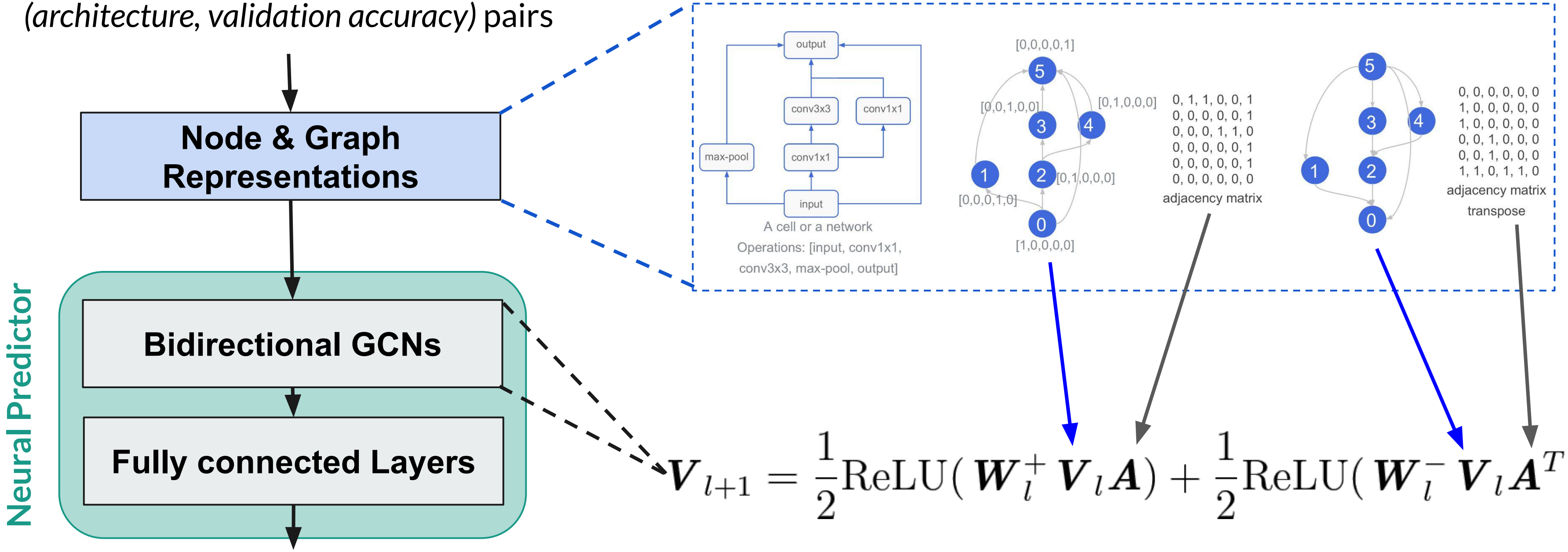
Fig. 1: Building (top) and applyi

Graph Convolutional Networks (GCNs)

Solution

Neural Predictor: Modeling by GCNs

Input as
(architecture, validation accuracy) pairs



$$\boldsymbol{V}_{l+1} = \frac{1}{2}\mathrm{ReLU}(\boldsymbol{W}_l^{+}\boldsymbol{V}_l\boldsymbol{A}) + \frac{1}{2}\mathrm{ReLU}(\boldsymbol{W}_l^{-}\boldsymbol{V}_l\boldsymbol{A}^T)$$

predicted accuracy *(e.g., 76.2)*

Solution

Neural Predictor

Step 2: Quality Prediction

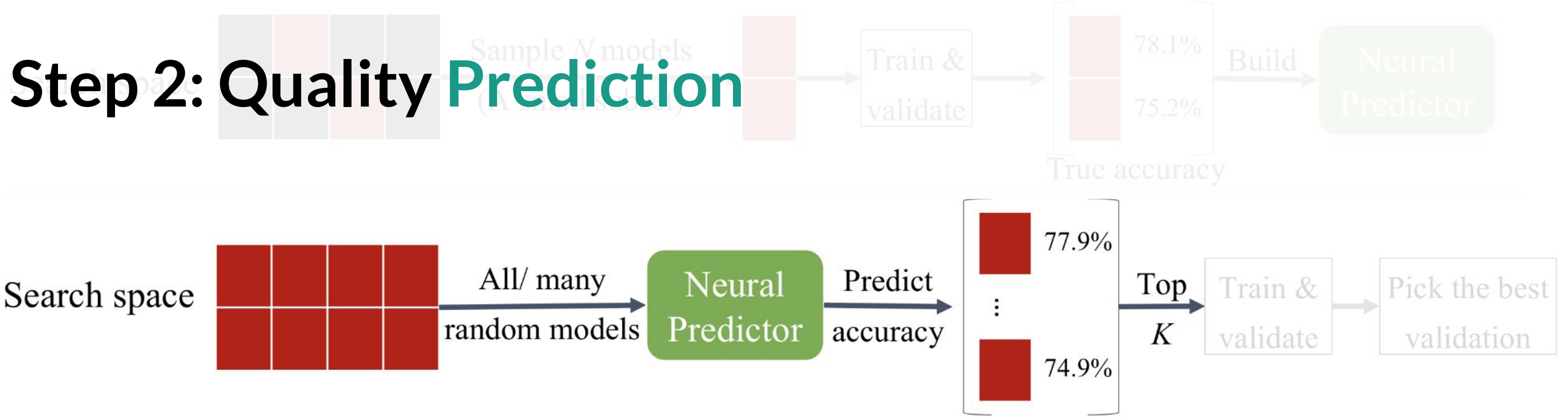


Fig. 1: Building (top) and applying (bottom) the Neural Predictor.

Solution

Neural Predictor

Step 3: Final Validation

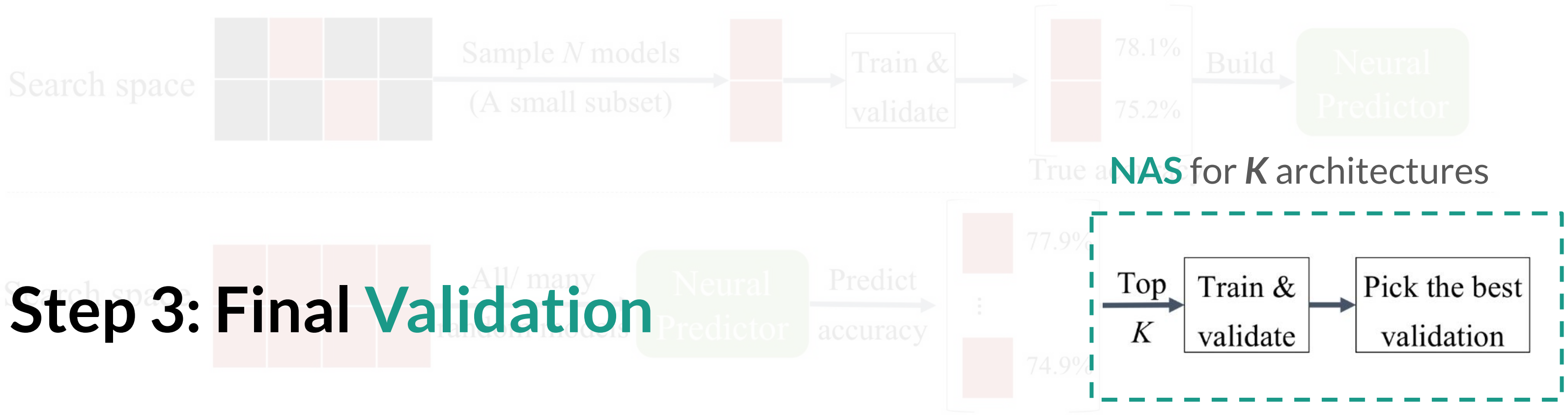
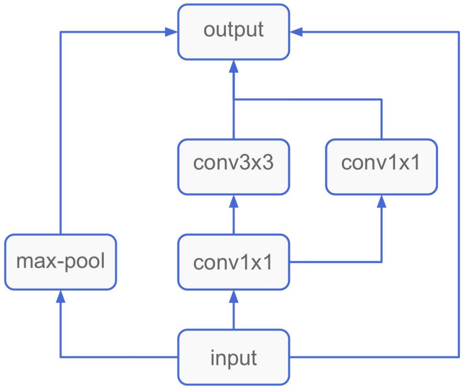


Fig. 1: Building (top) and applying (bottom) the Neural Predictor.

Contents

Replicated Experiments

Search Spaces / Datasets



NAS-Bench-101

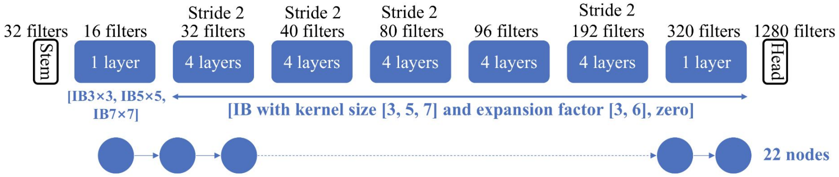
NAS-Bench-101 (CIFAR-10)

- **Size**: 423,624 models
- **Graph structures**: DAG with up to **7** nodes
- **5 Operations:** Input, Output, Conv1x1, Conv3x3, MaxPool3x3

ProxylessNAS (ImageNet)

- **Size**: 6.64 x 10^{17} models
- **Graph structures**: Linear graph with up to **22** nodes
- **7 Operations:** Input, Output, IB3x3-M, IB5x5-M, IB7x7-M, Skip (i.e., zero), and M = {3, 6}

ProxylessNAS



IB = MobileNetv2-based Inverted Bottleneck

Replicated Experiments

Evaluation Methods

Metrics for Neural Predictor

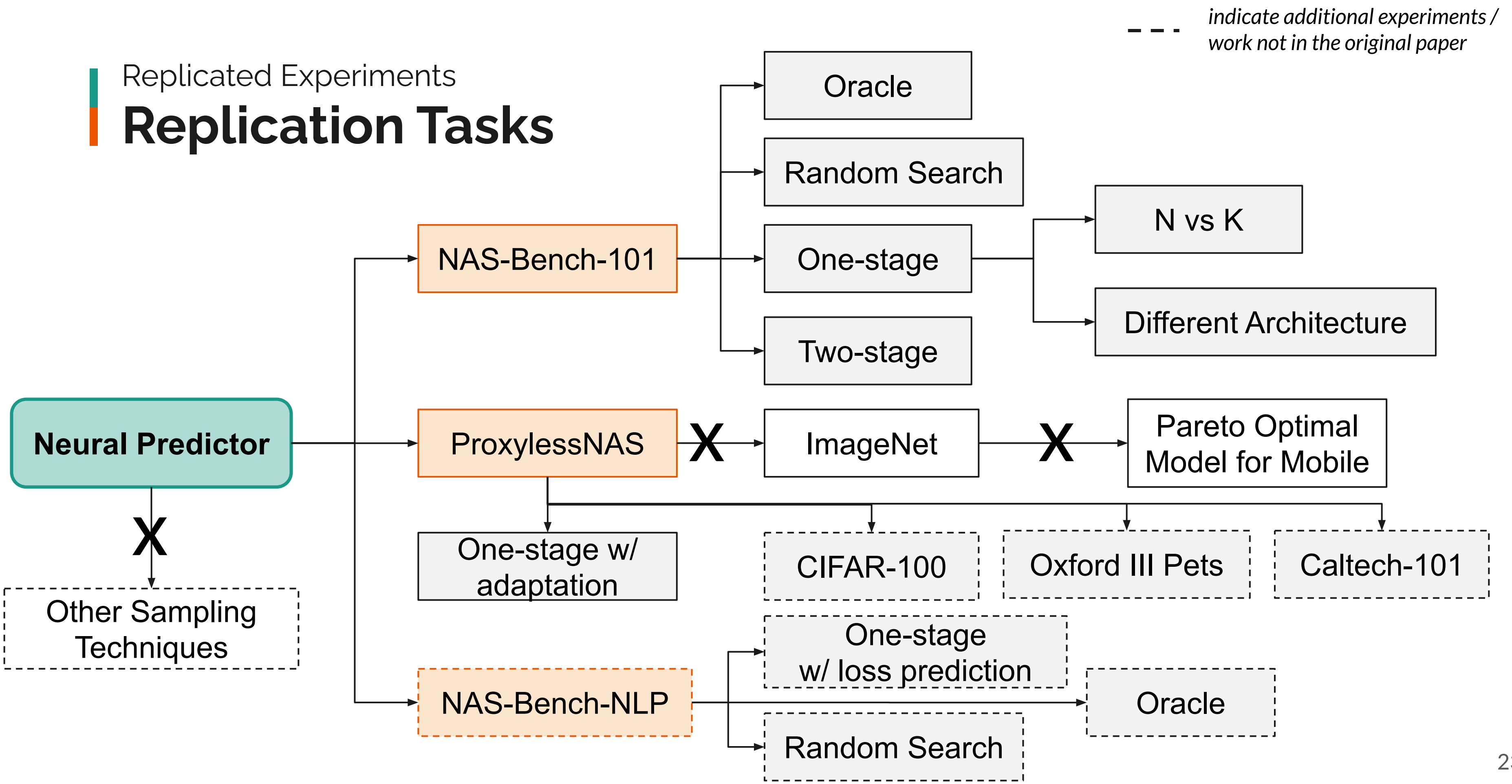
- **MSE** for training and testing
    - Lower is Better
- **Kendall rank** correlation coefficient for evaluating predicted accuracy
- R^2 **score** for evaluating predicted accuracy
    - Higher is Better

Metrics for Candidate Architectures

- **Accuracy** for architecture selection (validation) and performance report (test)
    - Higher is Better

Replicated Experiments

Replication Tasks



Replicated Experiments

Implementation Details

Tools / Libraries: Python 3, Tensorflow 2.5, and Keras

Platform: Ubuntu LTS 18.04 with a NVIDIA GTX GeForce 2080Ti GPU

Neural Predictor Default Settings:

- **# GCNs:** 3 for NAS-Bench-101 and 18 for ProxylessNAS
- **# FCs:** 1 for NAS-Bench-101 *(128 units)* and 2 for ProxylessNAS *(512 & 128 units)*
- **# GCN's node representation (channel):** 144 if N = 172 or else
- **Training parameters:** # Epochs ⇒ 300 and # Batch Size ⇒ 10
- **Optimizer**: Adam with initial learning rate of 0.0001 for regressor or 0.0002 for classifier
- **Learning Scheduler:** Cosine scheduling (i.e., Cosine annealing)
- **Weight decay:** 0.001
- **Dropout rate:** 0.1

N	D	N	D
43	48	172	144
86	72	334	210
129	96	860	320

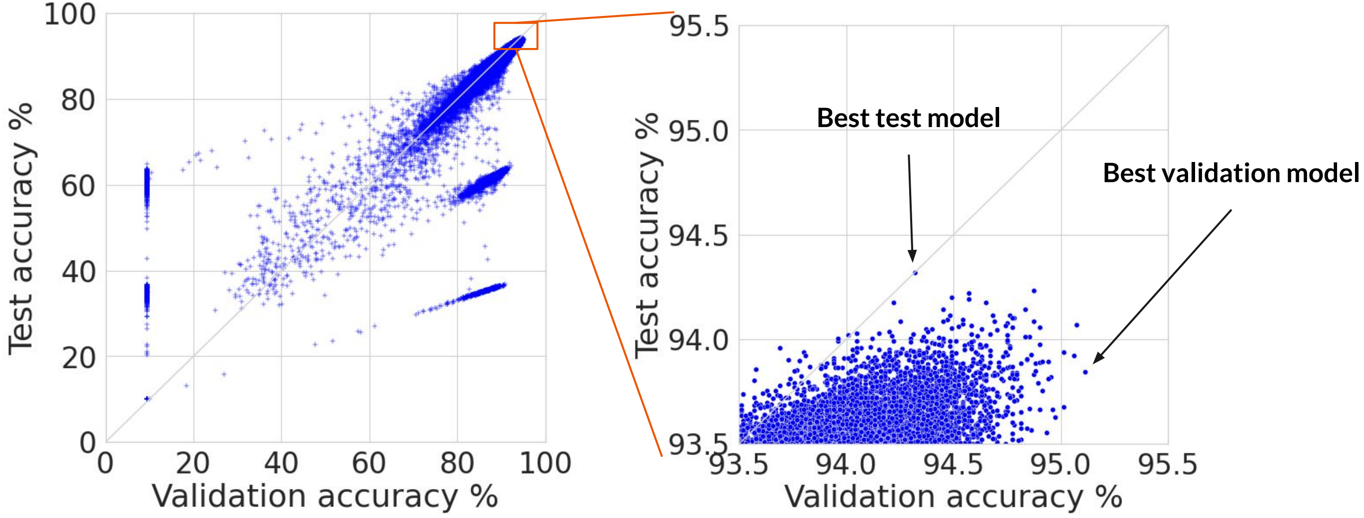
Default Hyperparameter Settings:

- **# Training Samples (N):** 172
- **# Validation Samples (K)**: 1 to 500-N

Contents

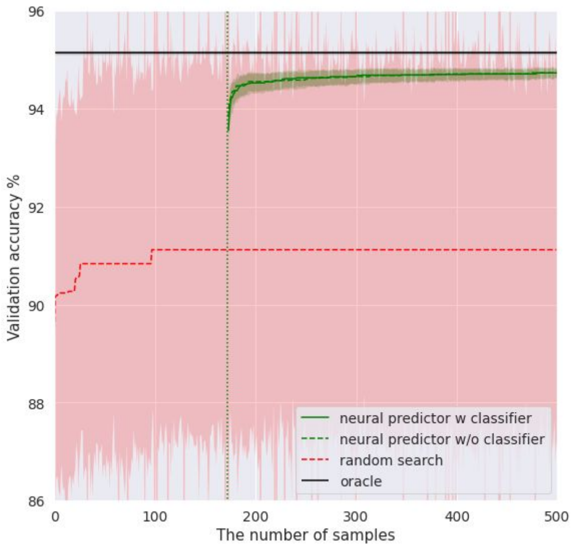
Results — NAS-Bench-101

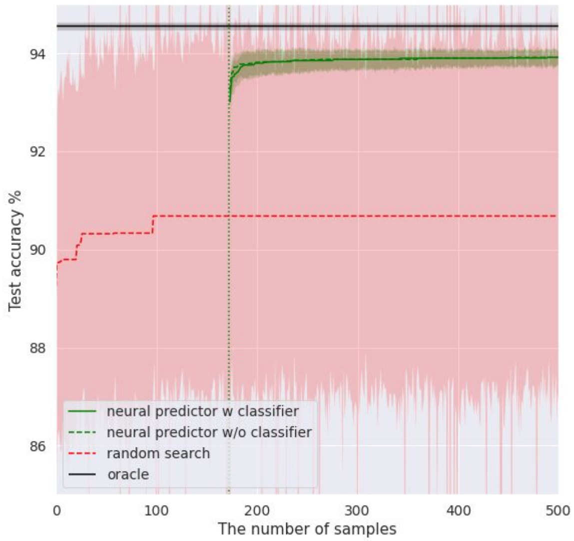
Understanding Oracle (Fig. 3)

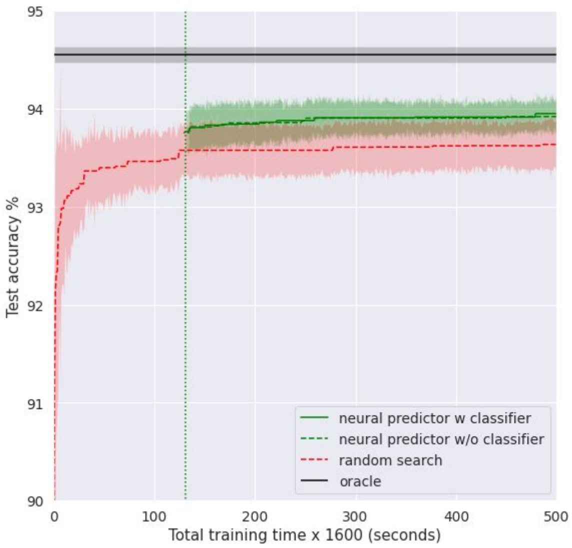


Results — NAS-Bench-101

Search Efficiency Comparison (Fig. 4)

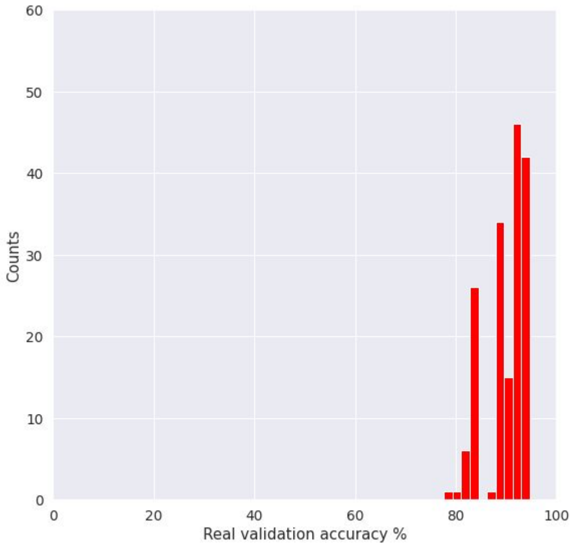


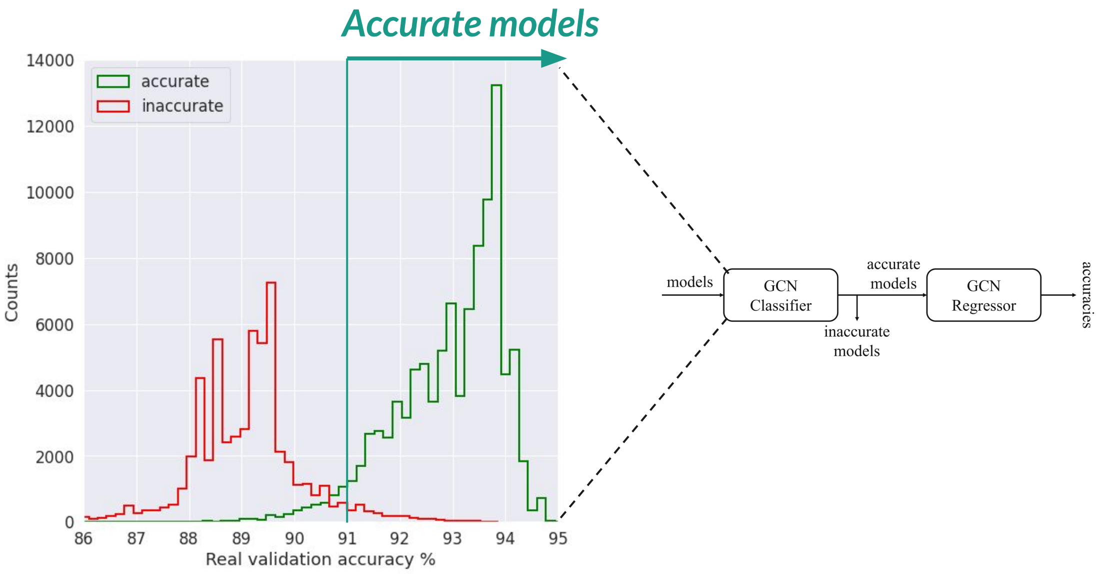




Results — NAS-Bench-101

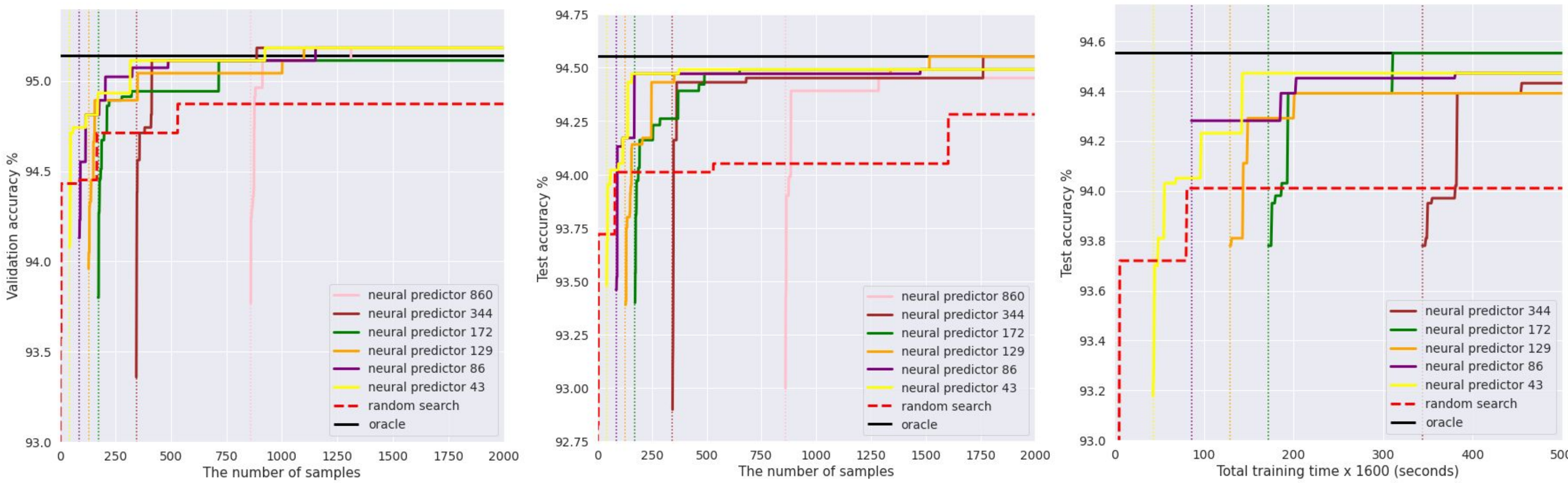
Two Stage Predictor (Fig. 6)





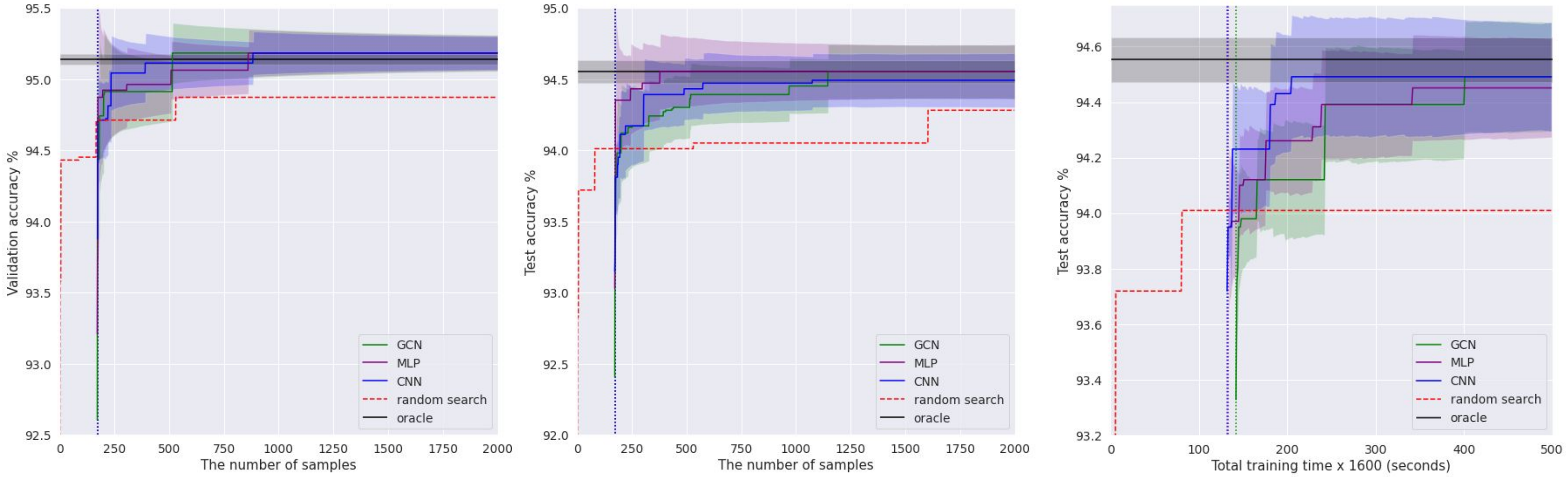
Results — NAS-Bench-101

Trade-off Analysis between N vs K (Fig. 7)



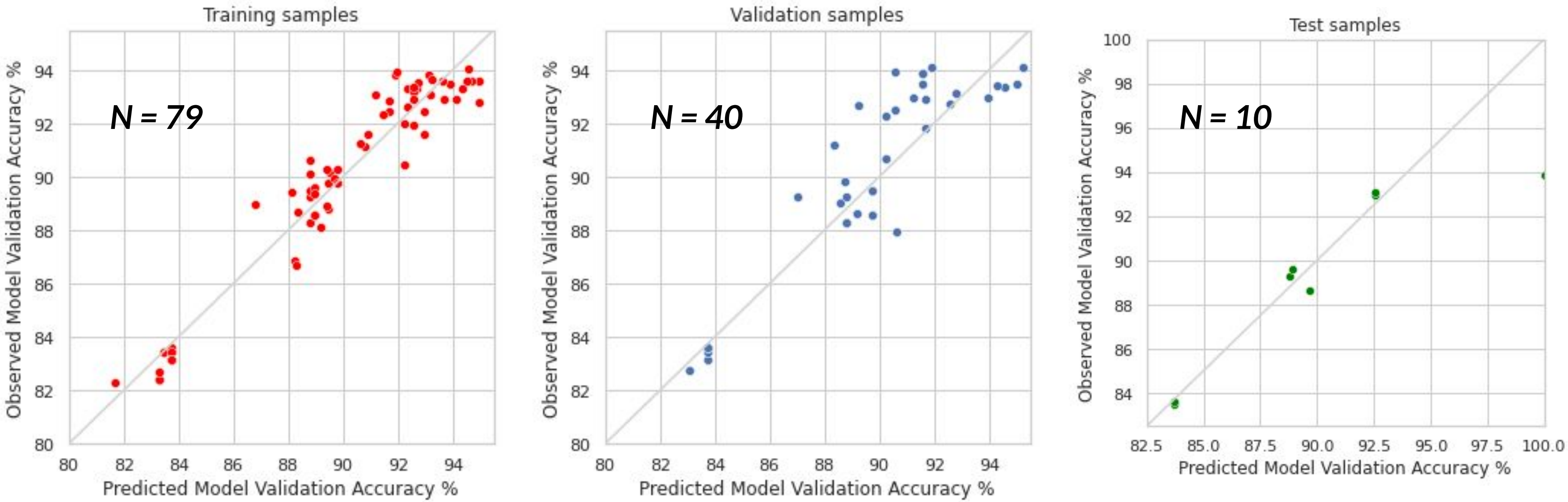
Results — NAS-Bench-101

Study on Different Architectures (Appendix Fig. 1)



Results — NAS-Bench-101

Performance of One-Stage Neural Predictor (Fig. 9)



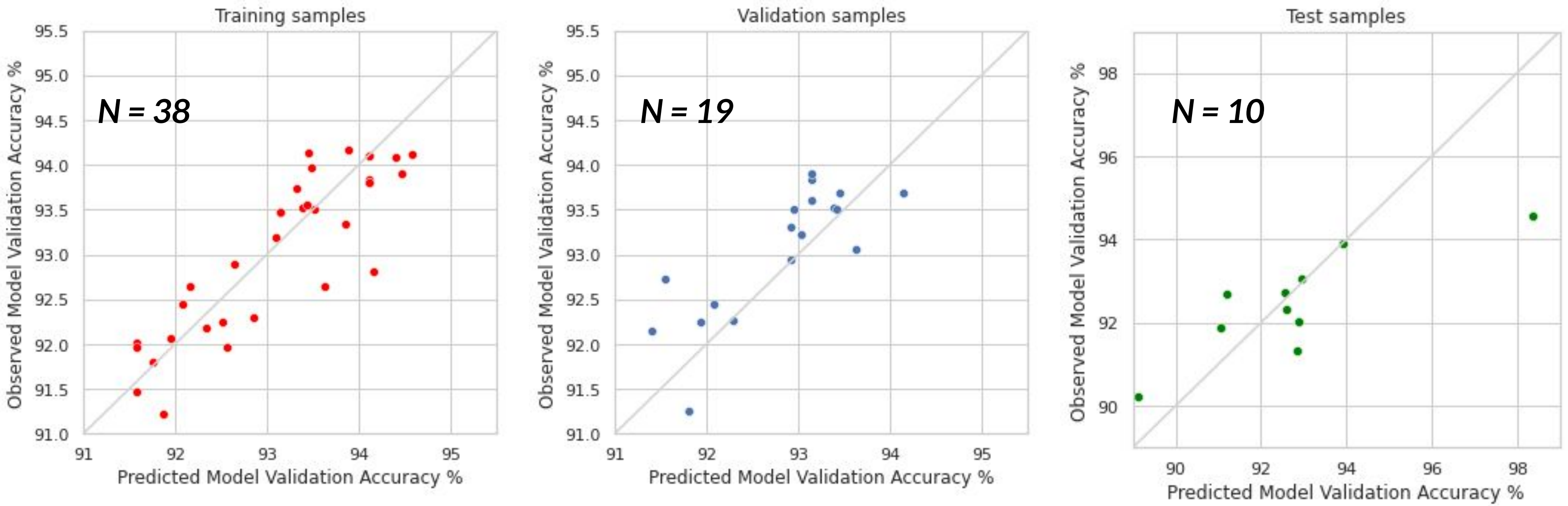
MSE: 0.826 , **Kendall Tua**: 0.799, **R^2**: 0.94842

MSE: 2.824 , **Kendall Tua**: 0.678, **R^2**: 0.79585

MSE: 3.982 , **Kendall Tua**: 0.837, **R^2**: 0.81668

Results — NAS-Bench-101

Performance of Two-Stage Neural Predictor (Fig. 9)



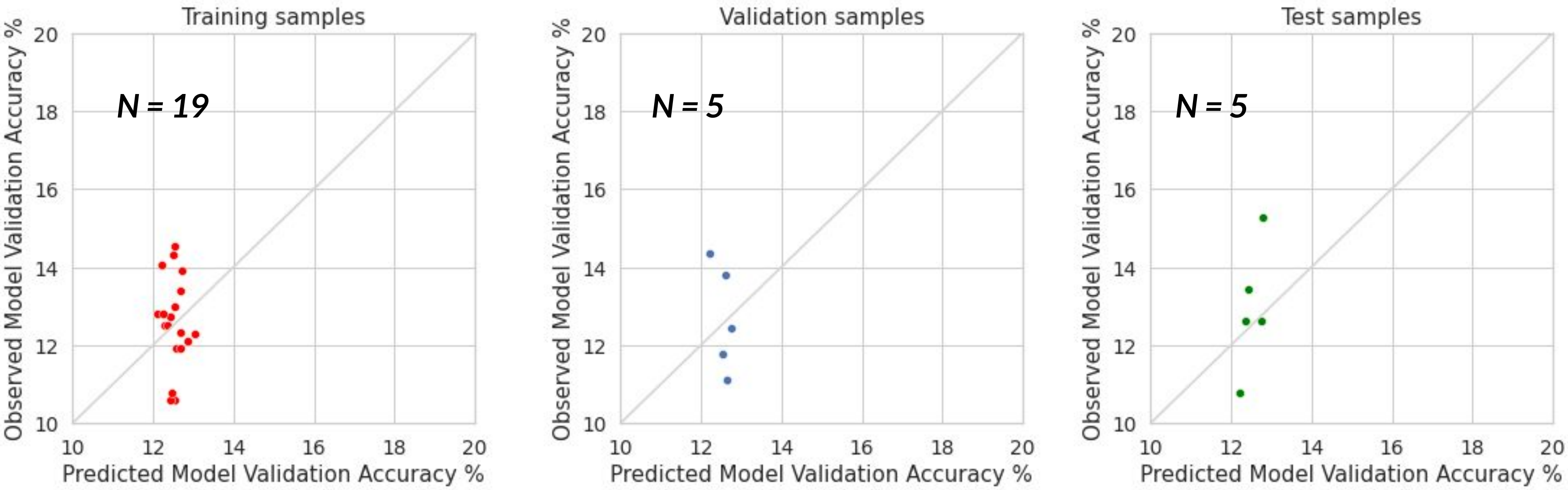
MSE: 0.234 , **Kendall Tua**: 0.681, **R^2**: 0.80341

MSE: 0.310 , **Kendall Tua**: 0.629, **R^2**: 0.50933

MSE: 4.005 , **Kendall Tua**: 0.182, **R^2**: -9.34282

Results — ProxylessNAS / CIFAR-100

Performance of Neural Predictor (Fig. 9)



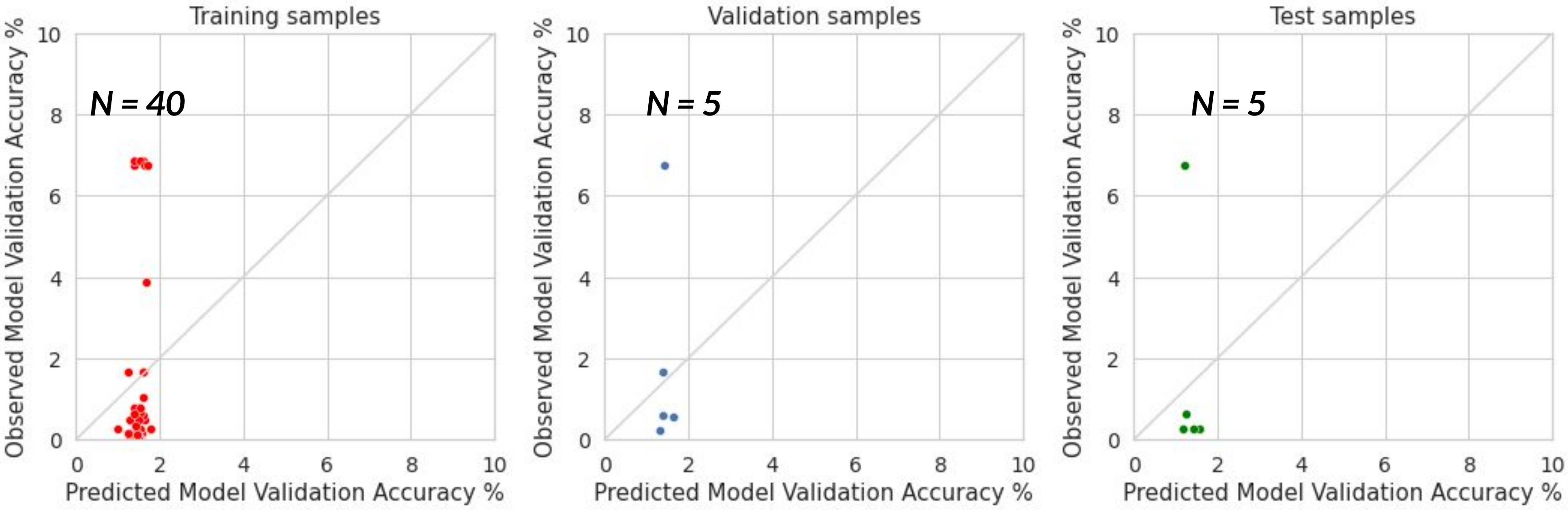
MSE: 1.335 , **Kendall Tua**: -0.053, **R^2**: -0.05951

MSE: 1.675 , **Kendall Tua**:0.0, **R^2**: -0.13005

MSE: 1.643 , **Kendall Tua**: 0.738, **R^2**: 0.21622

Results — ProxylessNAS / Caltech-101

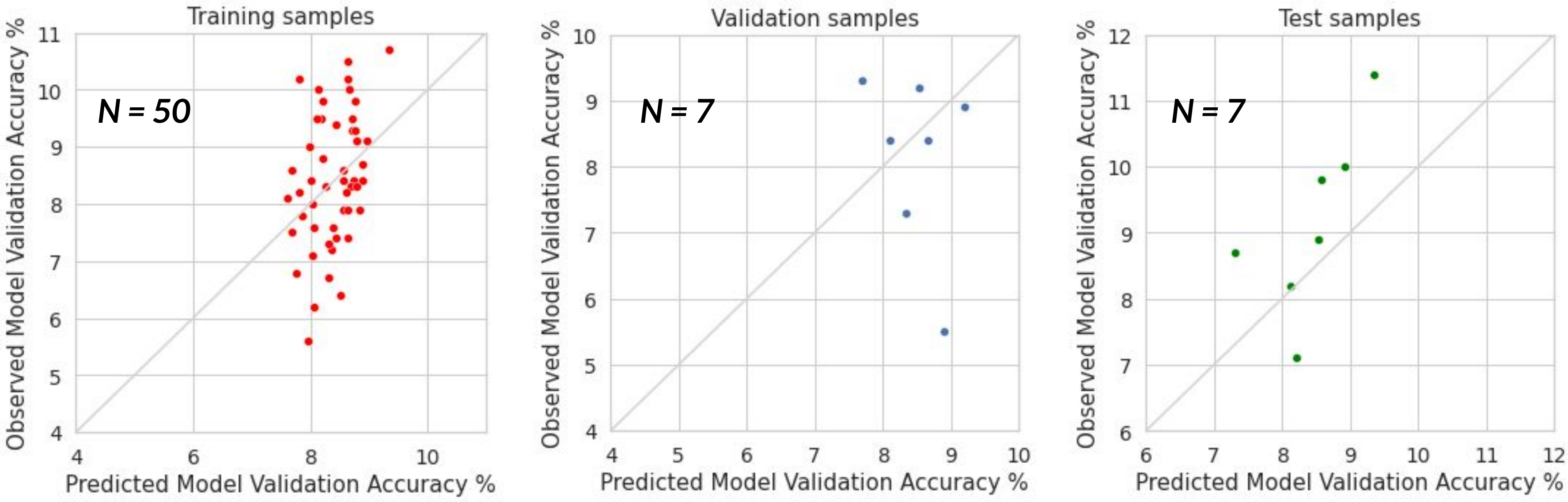
Performance of Neural Predictor (Fig. 9)



MSE: 5.295 , **Kendall Tua**: .235, $\mathbf{R^2}$: 0.02769 **MSE**: 6.319 , **Kendall Tua**: 0.200, $\mathbf{R^2}$: -0.05723 **MSE**: 7.023 , **Kendall Tua**: -0.358, $\mathbf{R^2}$: -0.07183

Results — ProxylessNAS / Oxford III Pet

Performance of Neural Predictor (Fig. 9)



MSE: 1.153 , **Kendall Tua**: 0.217, $\mathbf{R^2}$: 0.12045

MSE: 2.276, **Kendall Tua**: -0.292, $\mathbf{R^2}$: -0.46749

MSE: 1.461 , **Kendall Tua**: 0.714, $\mathbf{R^2}$: 0.11630

Contents

Discussion

Problems / Challenges

Unable to Utilize Parallelization

- Reduction in scale of the experiments (# total samples, training time).
- Dataset replacement. From ImageNet ⇒ smaller datasets.

Inaccessible Resources

- No latency prediction model was provided.
- Removing of Pareto front-based model finding.
- Unclear description of ProxylessNAS search space.

Exclusion of Other Sampling Techniques

- Search space is not available for access like a normal dataset.

Discussion

Conclusion

- We implement the *Neural Predictor* for training and predicting any given neural architectures based on NAS-Bench-101 and ProxylessNAS.
- We improve the Neural Predictor to work with ***NAS-Bench-NLP*** that requires the ***new preprocessing steps*** of RNN-based architectures and ***modification in the output layer*** for predicting loss instead of accuracy.
- We provide ***trained (architecture, validation accuracy) pairs*** for new datasets.
- We conduct almost the same experiments in the paper, except for their scales and end-to-end model finding with inference latency constraints due to the lack of resources.

Thank you!

20205642_**Patara Trirat**
20190754_**Guntitat Sawadwuthikul**

TEAM 3_**TH^2** / 2021.06.15
Source code: https://github.com/itouchz/Neural-Predictor-Tensorflow

Neural Predictor for Neural Architecture Search, ECCV'20

Wen, W., Liu, H., Chen, Y., Li, H., Bender, G., & Kindermans, P. J. (2020, August). Neural Predictor for Neural Architecture Search. In European Conference on Computer Vision (pp. 660-676). Springer, Cham.

References

Sampling-based NAS

[1] Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578 (2016)

[2] Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y.L., Tan, J., Le, Q.V., Kurakin, A.: Large-scale evolution of image classifiers. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 2902–2911. JMLR. org (2017)

[3] Dai, X., Zhang, P., Wu, B., Yin, H., Sun, F., Wang, Y., Dukhan, M., Hu, Y., Wu, Y., Jia, Y., et al.: Chamnet: Towards efficient network design through platformaware model adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11398–11407 (2019)

[4] Kandasamy, K., Neiswanger, W., Schneider, J., Poczos, B., Xing, E.P.: Neural architecture search with bayesian optimisation and optimal transport. In: Advances in neural information processing systems. pp. 2016–2025 (2018)

References

Weight Sharing-based NAS

[5] Bender, G., Kindermans, P.J., Zoph, B., Vasudevan, V., Le, Q.: Understanding and simplifying one-shot architecture search. In: International Conference on Machine Learning. pp. 549–558 (2018)

[6] Brock, A., Lim, T., Ritchie, J.M., Weston, N.: Smash: one-shot model architecture search through hypernetworks. arXiv preprint arXiv:1708.05344 (2017)

[7] Cai, H., Zhu, L., Han, S.: Proxylessnas: Direct neural architecture search on target task and hardware. arXiv preprint arXiv:1812.00332 (2018)

[8] Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055 (2018)

[9] Pham, H., Guan, M.Y., Zoph, B., Le, Q.V., Dean, J.: Efficient neural architecture search via parameter sharing. arXiv preprint arXiv:1802.03268 (2018)

References

Prediction-enhanced NAS

[10] Deng, B., Yan, J., Lin, D.: Peephole: Predicting network performance before training. arXiv preprint arXiv:1712.03351 (2017)

[11] Bender, G., Kindermans, P.J., Zoph, B., Vasudevan, V., Le, Q.: Understanding and simplifying one-shot architecture search. In: International Conference on Machine Learning. pp. 549–558 (2018)

[12] Baker, B., Gupta, O., Raskar, R., Naik, N.: Accelerating neural architecture search using performance prediction. arXiv preprint arXiv:1705.10823 (2017)

[13] Luo, R., Tian, F., Qin, T., Chen, E., Liu, T.Y.: Neural architecture optimization. In: Advances in neural information processing systems. pp. 7816–7827 (2018)

[14] Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L.J., Fei-Fei, L., Yuille, A., Huang, J., Murphy, K.: Progressive neural architecture search. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 19–34 (2018)

[15] Tang, Y., Wang, Y., Xu, Y., Chen, H., Shi, B., Xu, C., Xu, C., Tian, Q., Xu, C.: A semi-supervised assessor of neural architectures. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)