Neural Architecture Design and Robustness: A Dataset

Steffen Jung[1,2], Jovita Lukasik[1], Margret Keuper[1,2]
[1] Max Planck Institute for Informatics, Saarland Informatics Campus
[2] University of Siegen
{steffen.jung,jlukasik,keuper}@mpi-inf.mpg.de

Abstract

Finding architectures that are (more) robust against perturbations requires expensive evaluations. We introduce a database on neural architecture design and robustness evaluations to facilitate research in this direction. For this, we evaluate a whole neural architecture search space (NAS-Bench-201) on a range of common adversarial attacks and corruption types. We further present three exemplary use cases of this dataset, in which we (i) benchmark robustness measurements based on Jacobian and Hessian matrices for their robustness predictability, (ii) perform neural architecture search on robust accuracies, and (iii) provide an initial analysis of how architectural design choices affect robustness. We find that carefully crafting the topology of a network can have substantial impact on its robustness, where networks with the same parameter count range in mean adversarial robust accuracy from $20\% - 41\%$*. Code and data is available at* http://robustness.vision/.

1. Introduction

One factor of the ever-improving performance of deep neural networks is based on innovations in architecture design. However, human design of better performing architectures requires a huge amount of trial-and-error and a good intuition. Consequently, the automated search for new architectures (NAS) receives growing interest [5, 17]. Recently, NAS research is accompanied by the search for architectures that are robust against adversarial attacks and corruptions. This is important, since image classification networks can easily be fooled by small perturbations on the image data, of which some are even invisible for humans.

Robustness in NAS research combines the objective of high performing and robust architectures [9, 13]. However, there was no attempt so far to evaluate a full search space on robustness, but rather architectures in the wild. This paper is a first step towards closing this gap. We are the first to introduce a robustness dataset based on evaluating a *complete* NAS search space, such as to allow benchmarking neural architecture search approaches for the robustness of the found architectures, and investigate the effect of small architectural changes. This will facilitate better streamlined research on neural architecture design choices and their robustness. We evaluate all $6\,466$ unique pretrained architectures from the NAS-Bench-201 benchmark [5] on common adversarial attacks [4, 6, 11] and corruption types [7]. In summary we make the following contributions:

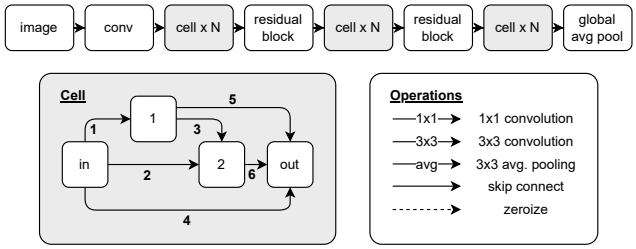

Figure 1. **(top)** Macro architecture. Gray highlighted cells differ between architectures, while the other components stay fixed. **(bottom)** Cell structure and the set of possible, predefined operations. (Figure adapted from [5])

- We present the first robustness dataset evaluating a complete NAS architectural search space.
- We present different use cases for this dataset; from training-free measurements for robustness to neural architecture search.
- Lastly, our dataset shows that carefully crafting architectures can substantially improve their robustness.

2. Dataset Generation

2.1. Architectures in NAS-Bench-201

NAS-Bench-201 [5] is a cell-based architecture search space. Each cell has in total 4 nodes and 6 edges. The nodes in this search space correspond to the architecture's feature maps and the edges represent the architecture's operations, which are chosen from the operation set $\mathcal{O} = \{1\times1 \text{ conv.}, 3\times3 \text{ conv.}, 3\times3 \text{ avg. pooling}, \text{skip}, \text{zero}\}$ (see

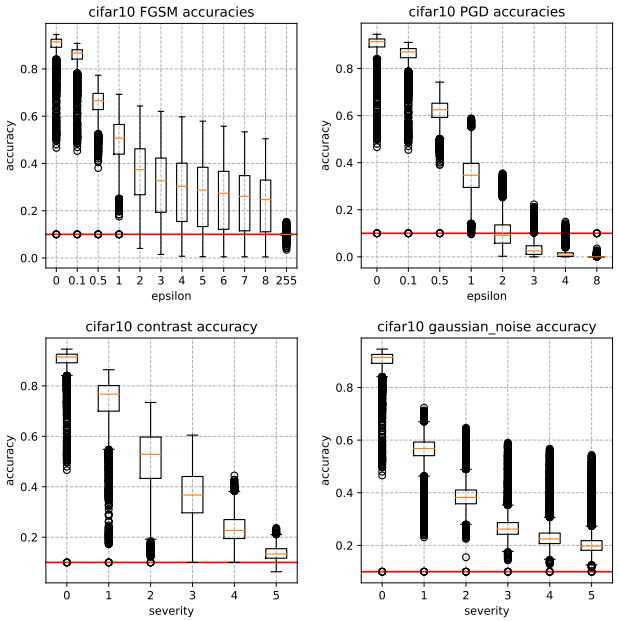

Figure 2. (**top**) Accuracy (FGSM [6] and PGD [11]) boxplots over all 6 466 unique architectures in NAS-Bench-201 for different perturbation magnitudes evaluated on CIFAR-10. (**bottom**) Accuracy boxplots for different corruption types at different severity levels evaluated on CIFAR-10-C. Red line corresponds to guessing.

Figure 1). This search space contains in total $5^6 = 15\,625$ architectures, from which only 6 466 are unique, since the operations skip and zero can cause isomorphic cells (zero stands for dropping the edge). Each architecture is trained on three different image datasets for 200 epochs: CIFAR-10 [10], CIFAR-100 [10] and ImageNet16-120 [3]. For our evaluations, we consider all unique architectures in the search space and test splits of the corresponding datasets. Hence, we evaluate $3 \cdot 6\,466 = 19\,398$ pretrained networks in total. In the following, we describe which evaluations we collect.

2.2. Robustness to Adversarial Attacks

FGSM FGSM [6] finds adversarial examples via

$$\tilde{x} = x + \epsilon \text{sign}(\Delta_x J(\theta, x, y)), \quad (1)$$

where $\tilde{x}$ is the adversarial example, x is the input image, y the corresponding label, ϵ the magnitude of the perturbation, and θ the network parameters. $J(\theta, x, y)$ is the loss function used to train the attacked network. Since attacks via FGSM can be evaluated fairly efficiently, we evaluate all architectures for $\epsilon \in E_{FGSM} = \{.1, .5, 1, 2, \ldots, 8, 255\}/255$, so for a total of $|E_{FGSM}| = 11$ times for each architecture.

PGD While FGSM perturbs the image in a single step of size ϵ, PGD [11] iteratively perturbs the image in smaller steps. As a result, PGD is more efficient in finding adversarial examples, but requires more computation time. Therefore, we find it sufficient to evaluate PGD for $\epsilon \in E_{PGD} = \{.1, .5, 1, 2, 3, 4, 8\}/255$ ($|E_{PGD}| = 7$).

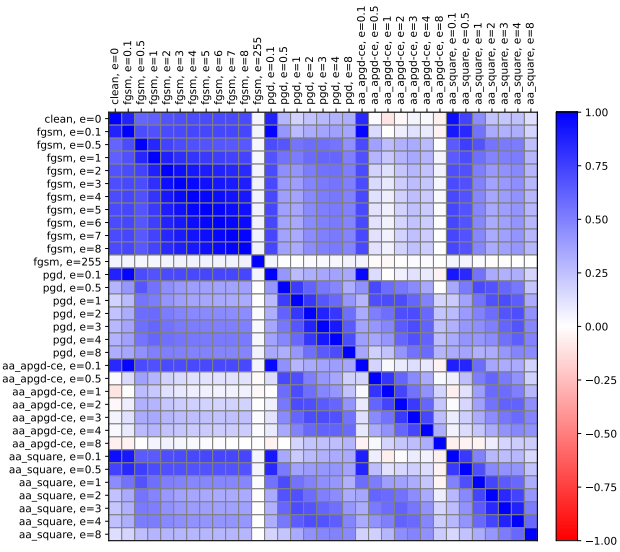

Figure 3. Kendall rank correlation between clean and robust accuracies on different attacks and magnitudes on CIFAR-10 for all unique architectures in NAS-Bench-201.

APGD AutoAttack [4] offers an adaptive version of PGD that reduces its step size over time. We kept the default number of attack iterations that is 100 [4] and choose $E_{APGD} = E_{PGD}$.

Square Attack In contrast to the before-mentioned attacks, Square Attack is a blackbox attack that has no access to the networks' gradients. It solves the following optimization problem using random search:

$$\min_{\tilde{x}} \{f_{y,\theta}(\tilde{x}) - \max_{k \neq y} f_{k,\theta}(\tilde{x})\}, \text{ s.t. } \|\tilde{x} - x\|_p \leq \epsilon, \quad (2)$$

where $f_{k,\theta}(\cdot)$ are the network predictions for class k given an image. We kept the default number of search iterations at 5 000 and choose $E_{Square} = E_{PGD}$.

Summary We collect (a) accuracy, (b) average prediction confidences, and (c) confusion matrices for each network and ϵ combination. Figure 2 shows aggregated evaluation results on before-mentioned attacks on CIFAR-10 w.r.t. accuracy. Growing gaps between mean and max accuracies indicate that architecture design has an impact on robust performances. Figure 3 depicts the correlation of ranking architectures based on different attack scenarios. While there is larger correlation within the same adversarial attack, there seem to be architectural distinctions for susceptibility to different attacks.

2.3. Robustness to Common Corruptions

To evaluate all unique NAS-Bench-201 [5] architectures on common corruptions, we evaluate them on the benchmark data provided by [7] (CIFAR10-C and CIFAR100-C). Both datasets are perturbed with a total of 15 corruptions at 5 severity levels (see Figure 16 in the Appendix for an example). The training procedure of NAS-Bench-201 only augments the training data with random flipping and random cropping. Hence, no influence should be expected of the training augmentation pipeline on the performance to those corruptions. We evaluate each network and collect (a) accuracy, (b) average prediction confidences, and (c) confusion matrices.

Summary Figure 2 depicts mean accuracies for different corruptions at increasing severity levels. Similar to adversarial attacks, the distribution of accuracies indicates towards architectural influence on robustness to common corruptions. Ranking architectures based on accuracy on different kinds of corruption is mostly uncorrelated and indicates towards a high diversity of sensitivity to different kinds of corruption based on architectural design.

3. Use Cases

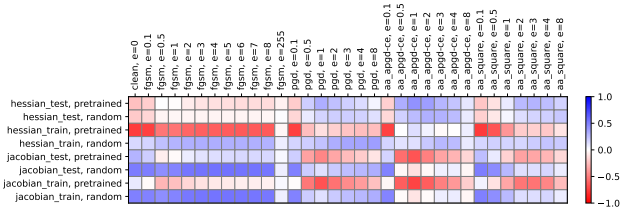

Figure 4. Kendall rank correlation between Jacobian- and Hessian-based robustness measurements computed on all unique NAS-Bench-201 architectures to rankings given by different adversarial attacks on CIFAR-10. Measurements are computed on randomly initialized and pretrained networks.

3.1. Training-Free Measurements for Robustness

Recent work [9, 13] finds high-scoring architectures, which are also adversarially robust, using training-free metrics based on Jacobian or loss landscape information of the neural networks. In this section, we evaluate these training-free gradient-based measurements with our dataset.

Jacobian To improve the robustness of neural architectures, [8] introduced a Jacobian-based regularization method with the goal to minimize the network's output change in case of perturbed input data. [8] shows, that the larger the Jacobian components, the larger is the output change and thus the more unstable is the neural network against perturbed input data. Therefore, the smaller the Frobenius norm of the Jacobian of a network, the more robust the network is supposed to be. We refer to [8] for more details. We compute the Frobenius norm on all 6 466 unique architectures and show the results in terms of ranking correlation to adversarial robustness in Figure 4, and observe that the Jacobian-based measurement correlates well with rankings after attacks by FGSM and smaller ϵ values for other attacks, which is not true anymore when ϵ increases, especially in the case of APGD.

Hessian [18] investigate the loss landscape of a regular neural network and robust neural network against adversarial attacks. [18] provide theoretical justification that the adversarial loss is highly correlated with the largest eigenvalue of the input Hessian matrix of the clean input data. Therefore, the eigenspectrum of the Hessian matrix of the regular network can be used for quantifying the robustness: large Hessian spectrum implies a sharp minimum, resulting in a more vulnerable neural network against adversarial attacks. We calculate the largest eigenvalues of all unique architectures using the Hessian approximation in [2]. These results are also shown in Figure 4. We can observe that the Hessian-based measurement behaves similarly to the Jacobian-based measurement.

Table 1. Neural Architecture Search on the clean test accuracy and the FGSM ($\epsilon = 1$) robust test accuracy for different state of the art methods on CIFAR-10 in the NAS-Bench-201 [5] search space (mean over 100 runs). Results are the mean accuracies of the best architectures found on different adversarial attacks and the mean accuracy over all corruptions and severity levels in CIFAR-10-C.

	Method	Test Accuracy ($\epsilon = 1.0$) Clean	FGSM	PDG	APGD	Squares	Clean
		CIFAR-10					CF-10-C
	Optimum	94.68	69.24	58.85	54.02	73.61	58.55
Clean	BANANAS [15]	94.21	64.25	41.10	18.62	68.69	55.52
	Local Search [16]	94.65	63.95	41.17	18.74	69.59	56.90
	Random Search [12]	94.22	63.38	40.09	17.84	68.40	55.60
	Regularized Evolution [14]	94.53	63.30	40.23	18.11	68.92	56.21
FGSM	BANANAS [15]	93.52	66.35	45.59	20.72	68.01	54.88
	Local Search [16]	93.86	69.10	48.27	23.18	69.47	56.57
	Random Search [12]	93.57	67.25	46.15	20.93	68.44	55.10
	Regularized Evolution [14]	93.77	68.82	47.99	22.59	69.20	56.11

3.2. NAS on Robustness

Table 1 shows the results of performing different state-of-the-art NAS algorithms on clean as well as FGSM ($\epsilon = 1$) robust accuracy in the NAS-Bench-201 [5] search space. Although clean accuracy is reduced, the overall robustness to all adversarial attacks improves when the search is performed on FGSM ($\epsilon = 1.0$) accuracy. Local Search achieves the best performance, which indicates that localized changes to an architecture design seem to be able to improve network robustness.

3.3. Analyzing the Effect of Architecture Design on Robustness

In Figure 5, we show the top-20 performing architectures (color-coded, one operation for each edge) with exactly 2

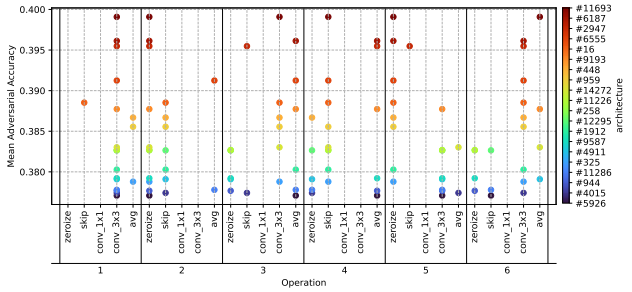

Figure 5. Top-20 architectures out of 408 that have exactly 2 times 3×3 convolutions and no 1×1 convolutions according to mean adversarial accuracy on CIFAR-10. The operation number (1-6) corresponds to the edge in the cell (see Figure 1).

times 3×3 convolutions and no 1×1 convolutions (hence, the same parameter count), according to the mean adversarial accuracy over all attacks as described in subsection 2.2 on CIFAR-10. It is interesting to see that there are no convolutions on edges 2 and 4, and additionally no dropping (operation zeroize) or skipping (operation skip-connect) of edge 1. In the case of edge 4, it seems that a single convolutional layer connecting input and output of the cell increases sensitivity of the network. Hence, most of the top-20 robust architectures stack convolutions (via edge 1, followed by either edge 3 or 5), from which we hypothesize that stacking convolution operations might improve robustness when designing architectures. At the same time, skipping input to output via edge 4 seems not to affect robustness negatively, as long as the input feature map is combined with stacked convolutions. Further analyses can be found in Appendix B. We find that optimizing architecture design can have a substantial impact on the robustness of a network. In this setting, where networks have *the same parameter count*, we can see a large range of mean adversarial accuracies $[0.21, 0.4]$ showing the potential of doubling the robustness of a network by carefully crafting its topology. Important to note here is that this is a first observation, which can be made by using our provided dataset. This observation functions as a motivation for how this dataset can be used to analyze robustness in combination with architecture design.

4. Conclusion

We introduce a dataset for neural architecture design and robustness to provide the research community with more resources for analyzing what constitutes robust networks. We evaluated *all* 6 466 unique architectures from the NAS-Bench-201 benchmark against several adversarial attacks and common corruptions and presented three use cases for this dataset: First, benchmarking robustness measurements. Second, NAS on robust accuracies, which indeed finds more robust architectures for different adversarial attacks. And last, an initial analysis of architectural design, where we showed that it is possible to improve robustness of networks with the same number of parameters by carefully designing their topology.

References

[1] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *ECCV*, 2020. 10

[2] A. Chatzimichailidis, J. Keuper, F.-J. Pfreundt, and N. R. Gauger. Gradvis: Visualization and second order analysis of optimization surfaces during the training of deep neural networks. In *Workshop on Machine Learning in High Performance Computing Environments, MLHPC@SC*, 2019. 3

[3] P. Chrabaszcz, I. Loshchilov, and F. Hutter. A downsampled variant of imagenet as an alternative to the CIFAR datasets. *CoRR*, abs/1707.08819, 2017. 2, 5

[4] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020. 1, 2, 10

[5] Xuanyi Dong and Yi Yang. Nas-bench-201: Extending the scope of reproducible neural architecture search. In *ICLR*, 2020. 1, 3, 5, 6, 14

[6] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 1, 2, 10

[7] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019. 1, 3, 10

[8] J. Hoffman, D. A. Roberts, and S. Yaida. Robust learning with jacobian regularization. *CoRR*, abs/1908.02729, 2019. 3

[9] R. Hosseini, X. Yang, and P. Xie. DSRNA: differentiable search of robust neural architectures. In *CVPR*, 2021. 1, 3

[10] Alex Krizhevsky. Learning multiple layers of features from tiny images. In *Technical report*, 2009. 2, 5, 15

[11] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale, 2017. 1, 2, 10

[12] L. Li and A. Talwalkar. Random search and reproducibility for neural architecture search. In *UAI*, 2019. 3

[13] J. Mok, B. Na, H. Choe, and S. Yoon. Advrush: Searching for adversarially robust neural architectures. In *ICCV*, 2021. 1, 3

[14] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le. Regularized evolution for image classifier architecture search. In *AAAI*, 2019. 3

[15] C. White, W. Neiswanger, and Y. Savani. Bananas: Bayesian optimization with neural architectures for neural architecture search. In *AAAI*, 2021. 3

[16] C. White, S. Nolen, and Y. Savani. Exploring the loss landscape in neural architecture search. In *UAI*, 2021. 3

[17] C. Ying, A. Klein, E. Christiansen, E. Real, K. Murphy, and F. Hutter. Nas-bench-101: Towards reproducible neural architecture search. In *ICML*, 2019. 1

[18] P. Zhao, P. Chen, P. Das, K. N. Ramamurthy, and X. Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. In *ICLR*, 2020. 3