Exploiting Model Compressibility in Zero-Cost NAS

Lichuan Xiang[1] , Łukasz Dudziak[2], Youyang Sha[1], Hongkai Wen[1,2*]
[1]University of Warwick [2] Samsung AI Center, Cambridge

Motivation

Inspired by the recent pruning at initialisation literature, “zero-cost" neural architecture search (NAS) methods [1] have demonstrated promising performance in effectively finding neural architectures with high accuracy at negligible search cost. Despite the different zero-cost metrics used, the key idea of those approaches is to sum over the saliency metrics of all parameters within the model as the overall ***zero-cost score*** for the evaluated architecture. However, this is often *biased*, evidenced by the limited correlations between the scores and the models’ final accuracy. Although the exact reason behind this remains an open question, if we take another look at the pruning process of a given model, we find an interesting discrepancy between the drop of model accuracy and its zero-cost scores as more parameters being removed (see Fig. 1). This means in NAS we will get architectures with similar accuracy but potentially different zero-cost scores, leading to inferior performance. Therefore, in this work we exploit the model compressibility to improve the correlations between zero-cost scores and ground truth accuracy.

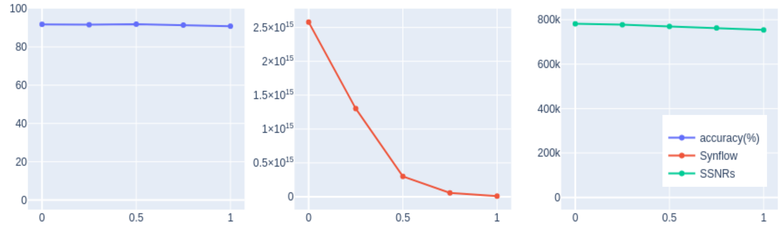

Figure 1. During the pruning process of the VGG-16 model (CIFAR-10), its zero-cost (`synflow`) score drops significantly before accuracy starting to decrease.

Our Solution

In pruning at initialization literature, the *saliency* of a parameter θ can be viewed as a signal that measures how much the parameter contributes to the final model accuracy. In particular, the following two saliency metrics (`grasp` and `synflow`) measure the second and first order information within the architecture, and have been widely adopted by recent zero-cost NAS approaches:

$$\texttt{grasp}: \mathcal{S}_p(\theta) = -(H\frac{\partial \mathcal{L}}{\partial \theta}) \odot \theta, \qquad \texttt{synflow}: \mathcal{S}_p(\theta) = \frac{\partial \mathcal{L}}{\partial \theta} \odot \theta$$

Where L is the loss function of a neural network with parameters θ, H is the Hessian, S_p is the per-parameter saliency and $\odot$ is the Hadamard product.

Therefore, as in [1], often the overall saliency of a given model is calculated by summing over the signals produced by all its parameters:

$$\mathcal{S}_n = \sum_i^N \mathcal{S}_p(\theta)_i$$

which is considered to be the zero-cost score of this model, and in the ideal case it should track the model accuracy well.

However, as shown by existing work [2], for a given model if we start to prune its parameters, those with lower saliency S_p are more likely to be dropped, while the model accuracy won’t be affected much. In that sense, for two architectures with similar overall saliency strength S_n, the one whose parameters have more diverse saliency signals should have a better chance to be pruned without harming its accuracy, i.e., higher compressibility. In the NAS context, this means it is likely that there exists other models which have lower overall saliency, i.e., zero-cost scores, but with similar ground truth accuracy, indicating the potential mismatch between the zero-cost scores and model accuracy. From a signal processing perspective, this indicates that the original S_n is not robust to background noise. To alleviate that we consider a new metric, the signal to noise ratio of saliency (SSNR), which is defined as follows:

$$S_{n/\sigma} = \frac{S_n}{\sigma}$$

where we normalize the overall saliency score S_n with the standard deviation of the per-parameter saliency signals within the network:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(S_p - \frac{S_n}{N})}$$

As shown in Fig. 1, compared to the saliency sum metric S_n, this overall SSNR ($S_{n/\sigma}$) shows much stronger correlation with the model accuracy.

In addition, as suggested by the conservation laws of synaptic saliency in [2], parameters from larger layers tend to have lower saliency scores, which are often under-estimated in terms of their contributions to the model accuracy. Therefore, we further modify the above SSNR metrics to a layer-wise form, to account for such diversity across layers. In particular for a given layer l, we break the previous overall S_n into a layer-wise vector $[S_n^1, \dots, S_n^l]$, and calculate the standard deviation for each layer as $[\sigma^1, \dots, \sigma^l]$. Then the final zero-cost score S of a model is defined as the sum of the SSNR for each layer:

$$S = \sum_{l=1}^{L}(\frac{S_n^1}{\sigma^l})$$

Preliminary Results

To verify the effectiveness of our SSNR based metric, we compute both the overall SSNR and layer-wise SSNR scores for all models in NAS-Bench-201 benchmark [3]. We show the correlations between the scores and the ground truth accuracy by plotting the distributions of the models in the score-accuracy space:

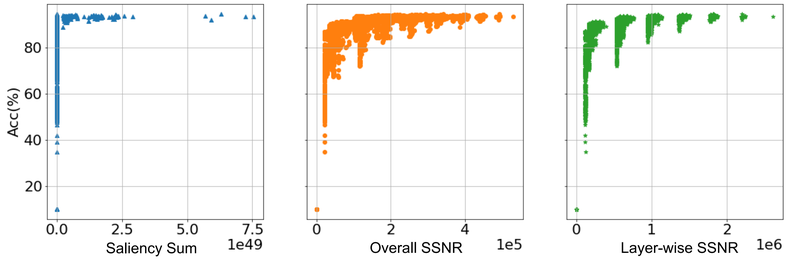

Figure 2. Distributions of models in score-accuracy space using different zero-cost metrics. (**Left**) Saliency Sum, (**Middle**) Overall SSNR, (**Right**) Layer-wise SSNR. All use `synflow` to calculate the per-parameter saliency.

We clearly see that our layer-wise SSNR metrics better track the model accuracy than the original saliency sum. Quantitively, the Spearman rank correlation between layer-wise SSNR and accuracy is 0.8178, while overall SSNR metric is 0.7785 and original zero-cost metric only achieves 0.7754. We further validate the robustness of the proposed zero-cost metrics on additional search spaces with different datasets, by running NASLib [4] with test size 5000. Results are shown below:

	NAS_BENCH_201		NAS_BENCH_301	Trasnbench101_micro		
	CIFAR-10	CIFAR-100	CIFAR-10	Jigsaw	class_object	class_scene
Synflow	0.5152	0.5615	0.1199	0.3311	0.4926	0.5408
SSNRs	0.5316	0.5252	0.2072	**0.3829**	0.4337	**0.5995**
Layerwise-SSNRs	**0.5549**	**0.5943**	**0.3392**	0.3807	**0.4356**	0.5794

Table 1. Kendall-tau correlation between different zero-cost metrics and model accuracy on different NAS benchmarks.

References

[1] Abdelfattah, M. S., Mehrotra, A., Dudziak, Ł., & Lane, N. D. (2020). Zero-Cost Proxies for Lightweight NAS. In International Conference on Learning Representations.
[2] Tanaka, H., Kunin, D., Yamins, D. L., & Ganguli, S. (2020). Pruning neural networks without any data by iteratively conserving synaptic flow. Advances in Neural Information Processing Systems, 33, 6377-6389.
[3] Dong, X., & Yang, Y. (2020). NAS-Bench-201: Extending the Scope of Reproducible Neural Architecture Search. ArXiv, abs/2001.00326.
[4] White, C., Zela, A., Ru, R., Liu, Y., & Hutter, F. (2021). How powerful are performance predictors in neural architecture search?. Advances in Neural Information Processing Systems, 34, 28454-28469.