ReLU is all you need for NASWOT

Prit Kanadiya[1, *] **Raghav Agarwal**[2, *] **Om Doiphode**[3, *] **Sandip Shingade**[4]

[1, 2, 3, 4]Veermata Jijabai Technological Institute
[*]Equal contribution.

Abstract From LeNet to Transformers, the design of neural network architectures has been central to the advancement of deep learning. Neural Architecture Search (NAS) aims to automate the design of neural network architectures, which traditionally is a time-consuming task and relies mainly on human intuition. Training-free heuristics such as NASWOT by Mellor et al. (2021) have emerged as efficient methods for estimating architecture performance without the need for training. However, NASWOT utilizes only the activation information obtained from the ReLU activation to score the network. While this works well for ReLU-dominated architectures such as CNNs, it does not utilize information from non-ReLU activations. This limits its applicability to domains such as natural language processing, which often rely on activations like sigmoid, tanh, and softmax along with ReLU. In this work, we build upon the NASWOT idea and generalize it to support a broad range of activation functions. Our generalized scoring function applies to any activation function whose output can be bounded within finite limits. We evaluate our method on the NAS-Bench-NLP and BERT Transformer benchmarks, expecting improvements over the original NASWOT baseline in terms of correlation with perplexity and BLEU score. Surprisingly, our results show minimal or no improvement, and even a decline in correlation for a few cases. Further experiments reveal a high correlation among the information captured by different activation functions, suggesting that diverse activations are encoding the same information. Moreover, ReLU alone appears to retain the most predictive information. These findings indicate that NASWOT remains effective within a subset of architectures, and that ReLU alone may be sufficient for capturing the key characteristics relevant to training-free performance prediction in such cases. The code for reproducing our experiments and results is available at `https://github.com/PritK99/Generalized-NASWOT`.

1 Introduction

Performance estimation strategy Elsken et al. (2019) is a key component of NAS Zoph et al. (2018). Instead of fully or partially training architectures, which is computationally expensive, NAS can use heuristics to estimate performance without training. This is called as NAS without training. One of the state-of-the-art training-free score function is NASWOT Mellor et al. (2021), which demonstrates a strong correlation with actual performance on CNN-based benchmarks. It evaluates architectures using ReLU activation patterns at initialization, encoding them as binary codes and computing Hamming distances between them to measure the network's discriminative capacity. A kernel matrix is constructed from these distances, and the NASWOT score is the log-determinant of this matrix. A higher score indicates stronger input separation, and this serves as a heuristic for actual performance of the neural network.

One limitation of NASWOT is its dependence on the ReLU activation function. Domains such as natural language processing use architectures like RNNs, LSTMs, GRUs, and Transformers, which predominantly rely on activation functions such as sigmoid, tanh, and softmax. As a result, there is a need to generalize NASWOT to support a broad class of activation functions. In this paper, we generalize NASWOT to support a wider range of activation functions with bounded outputs. We evaluate our generalized method on NLP benchmarks and compare it to the original NASWOT

baseline. Surprisingly our experiments show little to no improvement and in some cases a decline in performance. To understand this outcome we conduct further analysis and find that ReLU alone is sufficient to capture the information used by NASWOT.

2 Methodology

We make some observations in the kernel matrix formula proposed by NASWOT Mellor et al. (2021). Each term in the similarity kernel has two major components, one is a constant N_A and the other is the hamming distance between binary activation codes. The constant N_A represents the maximum possible hamming distance between any two binary codes. This concept makes the kernel matrix a measure of similarity of the two binary codes. A natural and intuitive extension is to consider metrics which are most suitable to capture the similarity between activations of other types. We propose a generalized similarity term for the kernel matrix given by

$$k_{i,j}{}^{(l)} = m - f(z_i^{(l)}, z_j^{(l)}) \tag{1}$$

Here, m is the maximum possible value taken by the function $f(z_i, z_j)$ and f is the function which measures the distance between two activation codes z_i and z_j.

Our proposed algorithm then starts with considering a mini-batch of data $\mathcal{D} = \{x_i\}_{i=1}^N$ and generating activation codes z_i from the value of activation functions in a neural network at x_i by using a suitable mapping. This mapping is used to bound the activation values so that $f(z_i^{(l)}, z_j^{(l)})$ has finite range and a defined value of m. For instance, NASWOT performs binary thresholding for ReLU activation which shrinks the ReLU range from $[0, \infty]$ to $[0, 1]$ and makes the hamming distance function $f(z_i^{(l)}, z_j^{(l)})$ bounded to the range $[0, N_A]$. It also provides unique codes for the linear regions created by the ReLU splits. Similar mappings can be used for other ReLU variants such as LeakyReLU and GELU.

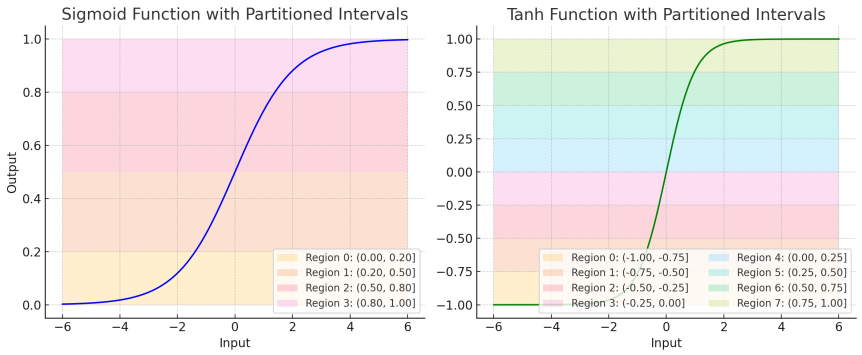

Figure 1: Partition regions for Sigmoid and Tanh

For smoother activation functions like sigmoid and tanh, we partition their output ranges into multiple discrete intervals to capture their nonlinear structure. To determine the optimal number of partitions, we use a subset of benchmark data as a validation set and experiment with 2, 4, 8, and 16 partitions. Additionally, we test the use of Euclidean distances on the raw activation values, effectively corresponding to infinite partitions. This validation set is separate from the test set used in our experiments. We observe that NASWOT performs best with 4 partitions for Sigmoid and 8 for Tanh. These partitions are illustrated in Figure 1. The Softmax activation outputs a probability distribution. Thus, we use the Jensen-Shannon divergence Lin (1991) to measure the similarity between two softmax activations.

Using the generalized similarity term from Equation (1), we compute the layer-wise kernel matrix $\mathbf{K}^{(l)}$ as:

$$\mathbf{K}^{(l)} = \begin{pmatrix} m - f(a_1^{(l)}, a_1^{(l)}) & \cdots & m - f(a_1^{(l)}, a_K^{(l)}) \\ \vdots & \ddots & \vdots \\ m - f(a_K^{(l)}, a_1^{(l)}) & \cdots & m - f(a_K^{(l)}, a_K^{(l)}) \end{pmatrix} \tag{2}$$

Here, m is the maximum value of the function f for the specific activation type, and $\mathbf{K}^{(l)}$ is the similarity kernel at layer l. We then aggregate the similarity kernels from all layers $\mathcal{L}$ to form the final kernel matrix K:

$$K = \sum_{l \in \mathcal{L}} \mathbf{K}^{(l)} \tag{3}$$

Finally, we define our overall score using the NASWOT formulation:

$$s = \log |K_H| \tag{4}$$

3 Experimental Results

We evaluate our generalized NASWOT scoring function on NAS-Bench-NLP Klyuchnikov et al. (2020) and the BERT-Transformer benchmark Serianni and Kalita (2023). We choose these NLP based benchmarks because they include architectures such as RNNs and Transformers which use a variety of activation functions.

NAS-Bench-NLP contains over 14,000 unique recurrent neural network architectures evaluated on the Penn Treebank dataset for language modeling. The architectures in this benchmark vary in RNN cell types, activation functions, and other hyperparameters. They use ReLU, sigmoid, and tanh activations, and their performance is measured using perplexity scores. In this case, a stronger negative correlation indicates better alignment with actual performance. This is because Perplexity is a loss-like metric in language modeling, and lower values indicate better performance. The BERT Transformer benchmark includes five hundred Transformer architectures pretrained on the OpenWebText corpus. These models are sampled from the FlexiBERT search space, which contains over ten million Transformer variants. The architectures use ReLU and softmax activations, and performance is assessed using GLUE scores. For this benchmark, a stronger positive correlation reflects better alignment with actual performance.

Table 1: Comparison across benchmarks and correlation metrics

	NAS-Bench-NLP (↓)			BERT Transformer (↑)		
Score Function	Kendall's Tau	Pearson	Spearman	Kendall's Tau	Pearson	Spearman
NASWOT	-0.27	**-0.24**	-0.43	**0.47**	**0.59**	**0.69**
Ours	**-0.31**	0.05	**-0.46**	0.21	0.15	0.32

For our evaluation, we randomly sample 1,000 architectures from NAS-Bench-NLP and use all 500 architectures from the BERT-Transformer benchmark. We report results using three correlation metrics which are Kendall Tau, Spearman rank correlation, and Pearson correlation. Table 1 presents the results obtained by the generalized NASWOT method on both benchmarks and compares them with the original NASWOT baseline. For NAS Bench NLP, including information from sigmoid and tanh activations results in a trivial improvement in Kendall Tau and Spearman correlation. However, Pearson correlation drops significantly, indicating that the relationship between the NASWOT score and model performance is no longer linear. For the BERT Transformer benchmark, including softmax information leads to a substantial 50% decline in performance across all three metrics.

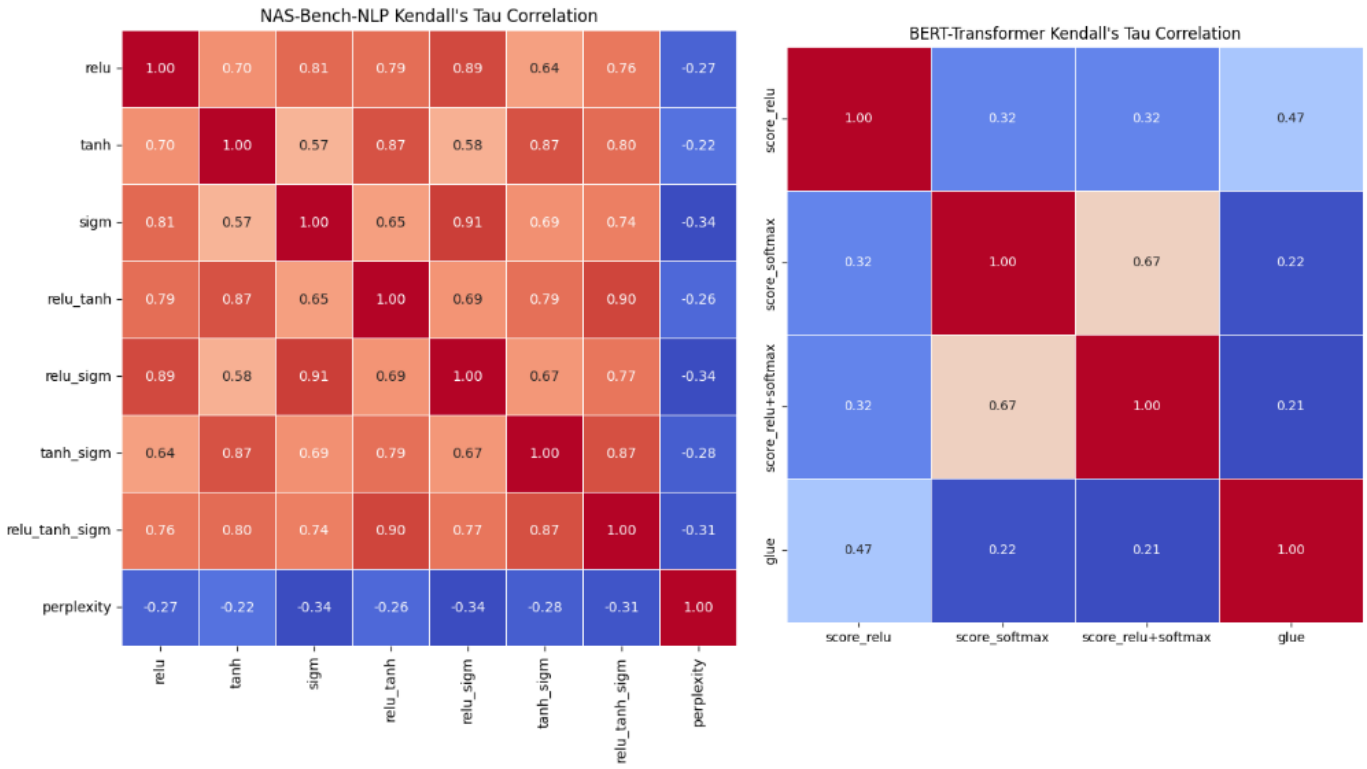

Figure 2: Kendall's Tau correlation for NAS-Bench-NLP and BERT Trasnformer

To understand the reasons behind the poor results, we analyzed the information captured by each activation type and their combinations. We then computed the Kendall's Tau correlation between each pair of combinations to assess the uniqueness of the information they captured. Figures 2 show the pairwise Kendall's Tau correlations for both benchmarks.

In the case of NAS-Bench-NLP, we observe a very high correlation among all combinations, indicating that they capture largely the same information. Similarly, for the BERT Transformer benchmark, the correlations are moderate, suggesting some degree of redundancy across combinations. The correlation patterns for Pearson and Spearman closely are similar to those of Kendall's Tau. Furthermore, in both benchmarks, ReLU alone achieves a strong correlation with ground truth. For BERT-Transformer, ReLU outperforms all other combinations. This suggests that the information obtained from ReLU activation is sufficient for the NASWOT score function.

To further support our analysis, we perform Principal Component Analysis (PCA) on the activation-based features to assess how information is distributed across components. We observe that the first principal component alone accounts for over 90% of the total variance, indicating that most of the information captured by the various activation combinations lies along a single dominant direction. This reinforces our claim that the activation combinations carry redundant information and ReLU is sufficient to capture the relevant information used by NASWOT.

4 Conclusion

Our attempt to generalize NASWOT shows that ReLU alone is sufficient to capture the information needed to measure the discriminative power of an untrained neural network. We suggest that future work explore other sources of information, such as the rank of Jacobian matrices, which may help improve or extend NASWOT. We hope our findings help guide further research on training-free performance prediction methods.

References

Elsken, T., Metzen, J. H., and Hutter, F. (2019). Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21.

Klyuchnikov, N., Trofimov, I., Artemova, E., Salnikov, M., Fedorov, M., and Burnaev, E. (2020). Nas-bench-nlp: Neural architecture search benchmark for natural language processing. *arXiv preprint arXiv:2006.07116*.

Lin, J. (1991). Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Mellor, J., Turner, R., Storkey, A., and Crowley, E. J. (2021). Neural architecture search without training. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*. PMLR.

Serianni, A. and Kalita, J. (2023). Training-free neural architecture search for rnns and transformers. *arXiv preprint arXiv:2306.00288*.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8697–8710.