A Study on Encodings for Neural Architecture Search

Colin White[1] Willie Neiswanger[2] Sam Nolen[1] Yash Savani[1]

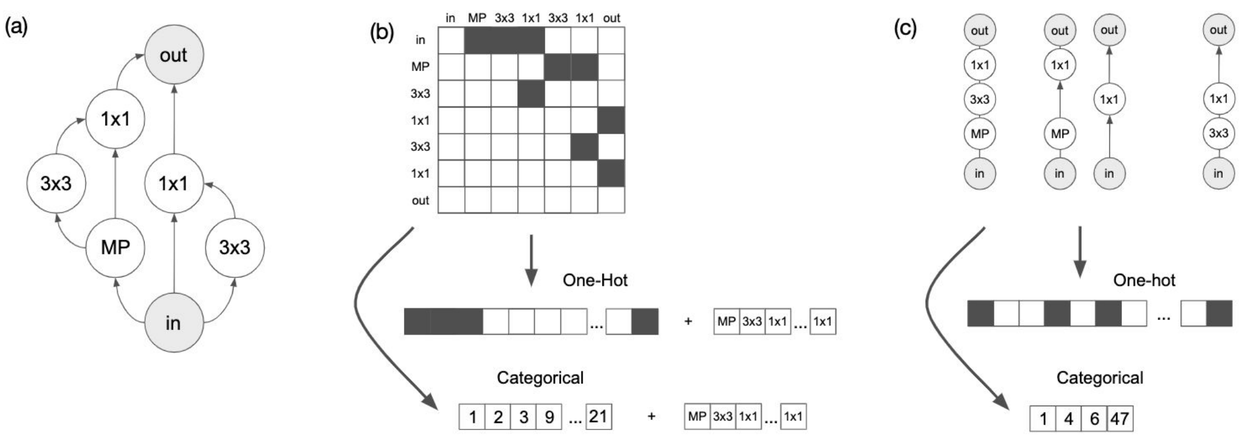
[1]Abacus.AI (*née* RealityEngines.AI) [2]Carnegie Mellon University

Introduction

- Neural architecture search (NAS) has been extensively studied in the past few years. A popular approach is to represent each neural architecture in the search space as a directed acyclic graph (DAG), and then search over all DAGs by encoding the adjacency matrix and list of operations as a set of hyperparameters. Recent work has demonstrated that even small changes to the way each architecture is encoded can have a significant effect on the performance of NAS algorithms [1, 2].
- In this work, we present the first formal study on the effect of architecture encodings for NAS, including a theoretical grounding and an empirical study. First we formally define architecture encodings and give a theoretical characterization on the scalability of the encodings we study. Then we identify the main encoding-dependent subroutines which NAS algorithms employ, running experiments to show which encodings work best with each subroutine for many popular algorithms. The experiments act as an ablation study for prior work, disentangling the algorithmic and encoding-based contributions, as well as a guideline for future work. Our results demonstrate that NAS encodings are an important design decision which can have significant impact on overall performance.

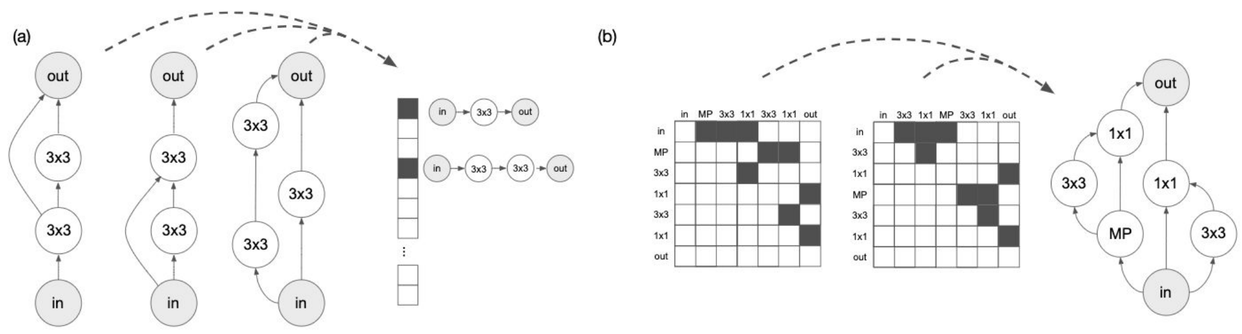
Encodings for NAS

We define a neural architecture encoding as an integer d and a multifunction e: $A \to \boldsymbol{R}^d$ from a set of neural architectures A to a d-dimensional Euclidean space $\boldsymbol{R}^d$. The encoding is a fixed transformation, independent of the dataset.



We define eight different encodings based on two paradigms:

- Adjacency matrix based encodings, including one-hot, categorical, and continuous variants. See the above figure, part (b).
- Path based encodings, including one-hot, categorical, and continuous variants. See the above figure, part (c). The one-hot and continuous variants can be truncated to define two additional encodings.



- The path encoding is not one-to-one, as shown in the above figure, part (a). Three different architectures map to the same encoding.
- The adjacency matrix is a multi-function, as shown in the above figure, part (b). Two different encodings map to the same architecture.

The Scalability of Encodings

- In prior work, the one-hot path encoding was shown to be effective on smaller benchmark NAS datasets [2,3] but there have been questions of whether its exponential length allows it to perform well on larger search spaces.
- We show that the path encoding can be significantly truncated with little loss of information, while the adjacency encoding cannot be truncated.

Given a random neural architecture with k edges, n nodes, and r choices of operations on each node, let $b(k,x)$ denote the expected fraction of paths of size x or less.

Theorem 4.1. Given $10 \leq n \leq k \leq \frac{n(n-1)}{2}$, and $c > 3$, for $x > 2ec \cdot \frac{k}{n}$, $b(k,x) > 1 - c^{-x+1}$, and for $x < \frac{1}{2ec} \cdot \frac{k}{n}$, $b(k,x) < -2^{\frac{k}{2n}}$.

This means, for the purposes of NAS, truncating the path encoding to $r^{k/n}$ contains almost exactly the same information as the full path encoding, and it cannot be truncated any smaller.

Theorem 4.2. Given $n \leq k \leq \frac{n(n-1)}{2}$ and $1 \leq z \leq n(n-1)/2$, we have $P(z \in E_{n,k,r}) > \frac{2k}{n(n-1)}$.

This means, for the purposes of NAS, the adjacency matrix cannot be truncated even by one bit without losing a significant amount of information.

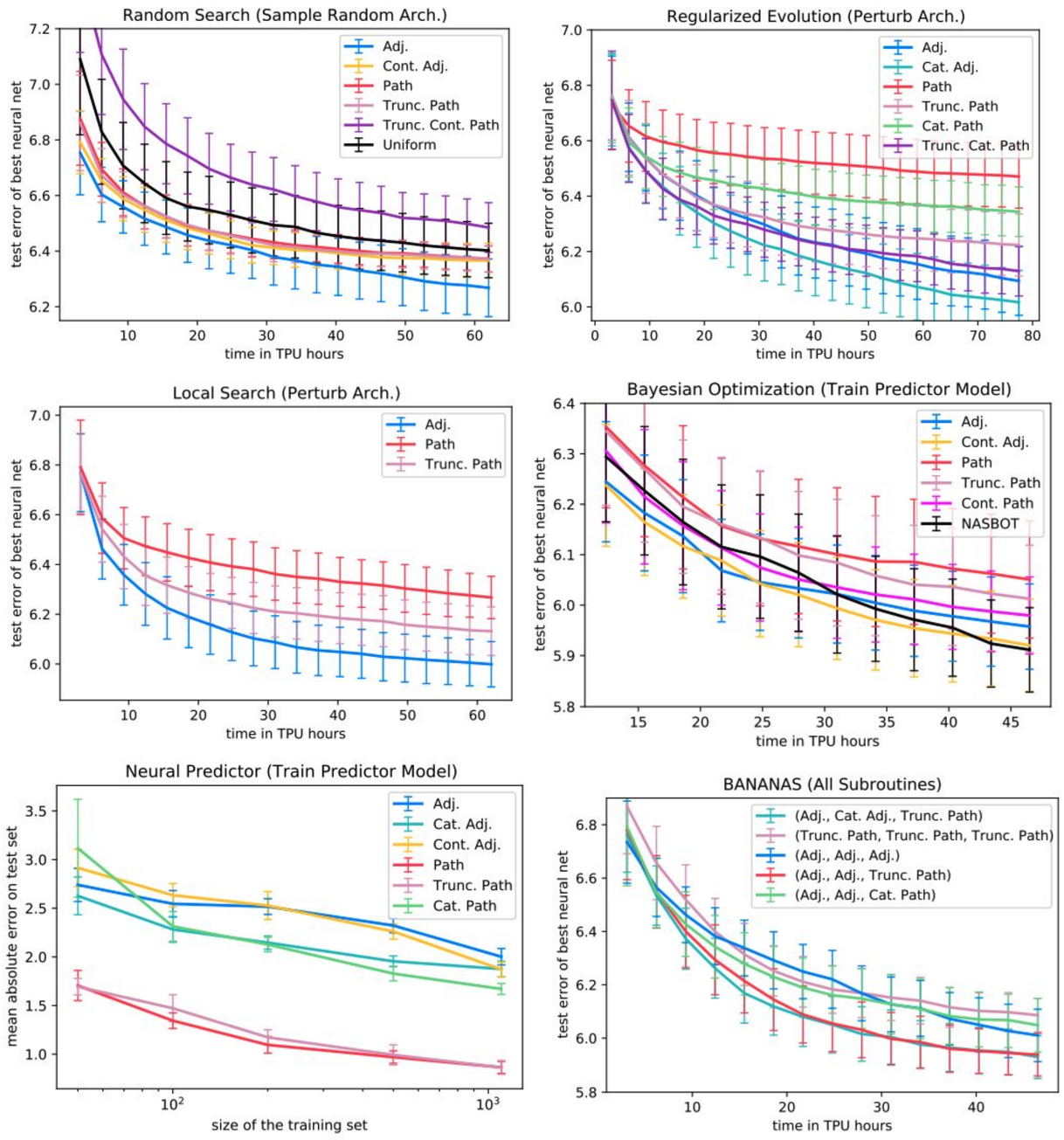
Conclusion

- We demonstrate that the choice of encoding is an important, nontrivial question that should be considered not only at the algorithmic level, but at the subroutine level.
- We give a theoretical grounding and characterize the scalability of NAS encodings.
- We give an experimental study of encodings for NAS, disentangling the algorithmic contributions from the encoding-based contributions of prior work, and laying out recommendations for the best encodings to use in future work.

Encoding Experiments

We split up our experiments based on three encoding-dependent subroutines. These three subroutines are the only encoding-dependent building blocks necessary for many NAS algorithms.

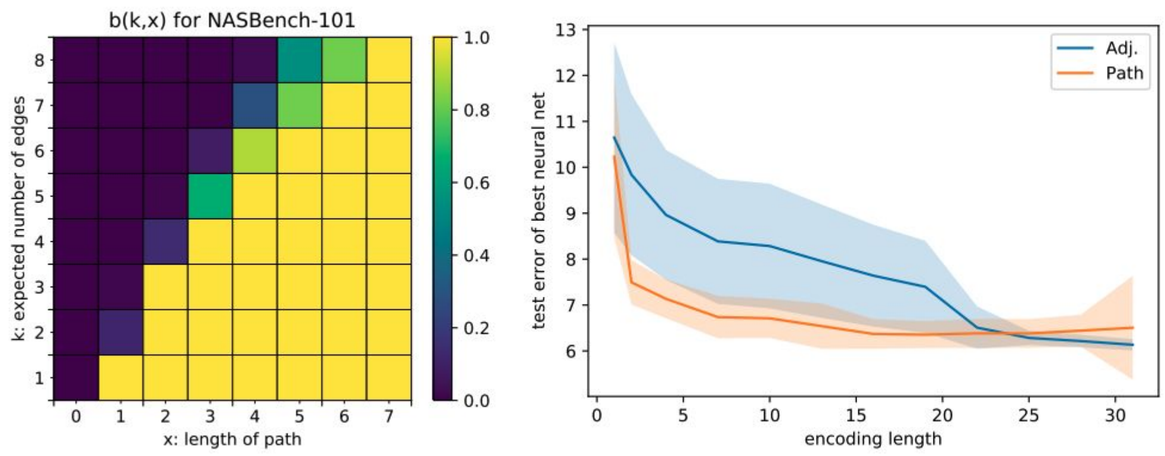
- Sample random architecture - draw an architecture randomly from the search space.
- Perturb architecture - make a small change to a given architecture.
- Train predictor model - train a model (e.g., GP or neural network) for prediction.



Next, we test the ability of a neural predictor to generalize to new search spaces, using one of four different encodings. We trained each neural predictor on a subset of NAS-Bench-101, and then tested on a disjoint test set of architectures.

Encoding	Validation error		Test error	
	Top 10 avg.	Top 1 avg.	Top 10 avg.	Top 1 avg.
Adjacency	**5.888**	**5.505**	**6.454**	**6.056**
Categorical Adjacency	7.589	6.191	8.155	7.086
Path	5.967	5.606	6.616	6.335
Truncated Path	6.082	5.644	6.712	6.452
Categorical Path	6.357	5.703	6.939	6.489
Truncated Categorical Path	6.339	5.895	6.918	6.766

Finally, we give evidence for the theoretical results on the scalability of encodings. We plot the empirical values of $b(k,x)$, and we plot the performance of the adjacency and path encodings as a function of the encoding length.



References

Full-length paper: https://arxiv.org/pdf/2007.04965.pdf

[1] Chris Ying, Aaron Klein, Esteban Real, Eric Christiansen, Kevin Murphy, and Frank Hutter. Nas-bench-101: Towards reproducible neural architecture search. ICML 2019.
[2] Colin White, Willie Neiswanger, and Yash Savani. Bananas: Bayesian optimization with neural architectures for neural architecture search. arXiv preprint arXiv:1910.11858, 2019.
[3] Chen Wei, Chuang Niu, Yiping Tang, and Jimin Liang. Npenas: Neural predictor guided evolution for neural architecture search. arXiv preprint arXiv:2003.12857, 2020.