Exploring Trade-offs in Computational Efficiency and Interpretability in NAS Structures

Jeanine Ohene-Agyei
University of Toronto
j.oheneagyei@mail.utoronto.ca

Abstract

Deep neural networks (DNNs) have become a central focus of research and development in several fields. Nonetheless, the increasing intricacy of these structures has presented difficulties not just in maximizing their computational effectiveness but also in their interpretability. Neural architecture search (NAS) can reduce the time it takes to design a DNN, however it can be computationally expensive and lacks the production of interpretable structures. Prior work aimed at resolving this issue has focused on image-based convolutional neural networks without much insight on the computational complexity or interpretability of other neural networks. I propose EcoNAS, a multi-objective evolutionary benchmark for feed forward, fully-connected DNNs that maximizes accuracy and interpretability, and minimizes computational complexity. This benchmark generates precomputed objective datasets and uses decision tree regression models to rapidly evaluate hundreds of architectures in a matter of seconds. I demonstrate the effectiveness of this benchmark to reduce computational complexity of DNNs, yet maintain high-performance and interpretability.[1]

1 Introduction

In pursuit of optimal model performance, deep neural networks (DNNs) have become a central focus of research and development in the fields of artificial intelligence and machine learning. From image recognition to natural language processing, DNNs have demonstrated unprecedented capabilities. However, the growing complexity of these architectures has led to challenges, not only in optimizing their computational efficiency, but their interpretability as well. Deep neural networks consist of several hidden layers each of varying size, and the more complex their task is, the more complex they may become. A proposed solution to this problem: neural architecture search (NAS), a technique used in machine learning to automate the ideal design of artificial neural networks (ANNs). NAS reduces the time it takes to design a DNN, and many ANNs can outperform hand-created DNNs.

Nonetheless, neural architecture search often requires large computational resources, due to its expensive training and evaluation phases, and the architectures produced by NAS are not optimized for interpretability. There has been great work in reducing the computational impact of NAS through

[1] The code and datasets used are available at https://github.com/jeanine5/UofT-ProjectX-2023

benchmarks, such the NAS-Benchmark-201 [13]. However, these works do not provide insight on reducing the computational complexity or increasing the interpretability of DNNs beyond convolutional neural networks (CNNs). Previous works such as [1, 2, 4, 5, 6, 12] create their own benchmarks to improve computational complexity *or* interpretability on convolutional DNNs. As other DNNs are still widely used for non-computer vision tasks, it is important to find a way to improve their computational efficiency, as well as their interpretability, for specific tasks in an efficient way. In this paper, I put forth the following contributions:

- I propose a multi-objective evolutionary benchmark, *EcoNAS*, for feed forward, fully-connected DNNs that maximizes accuracy and interpretability, and minimizes computational complexity.
- I develop evaluation datasets on trained DNNs and use decision trees to rapidly evaluate hundreds of similar architectures in seconds.
- I analyze the tradeoffs between each of the objectives on the best performing architectures through image classification datasets.

2 Background

2.1 Interpretability

Interpretability is the ability to explain or present in understandable terms to a human [9]. In machine learning, it is how one can explain or predict a model's decisions. However, interpretability is difficult to define mathematically and therefore, transforming this definition into a performance metric for NAS algorithms is even more challenging. For this, there are two proposed solutions. The first is to allow interactive features into the model that allows users to interpret the model as it trains. This solution is proposed by [9] and requires human input to select the meaningful neurons from each layer. This solution however becomes a less practical technique when working with an incredibly large network, as the human must pick neurons for possibly hundreds of layers. The second solution proposed by [1] is to modify the way we would define interpretability. The creators of XNAS coined the term *introspectability*: the extent to which the neural network *M* segregates the representations of distinct classes [1]. By using this definition, we can define introspectability mathematically as

$$\text{Introspectability}(M, X) = \frac{1}{\binom{N_c}{2}} \sum_{c=1}^{N_c} \sum_{c=k+1}^{N_c} D(\phi(c), \phi(k)), \quad [1]$$

where D(.,.) is the cosine distance between two vectored arguments and N_C is the number of classes in the classification task [1].

2.2 Multi-Objective Optimization

In practical applications, it is often necessary to trade off two or more objectives, such as the performance of a model and resource consumption or multiple loss functions [7]. In a multi-objective optimization problem, we are given M objectives $f(x) = \{f_1(x),..., f_m(x)\} \in R^M$, in which we must optimize each objective given some constraints on the input vector x [3]. When some objectives in $f(x)$ need to be *minimized* while others need to be *maximized*, we say they are conflicting. When objectives are conflicting, we must find a way to handle them. One method is to convert the multi-objective optimization into a single objective by introducing weight hyperparameters [3]. An ideal solution for conflicting objectives is to maintain Pareto-optimal solutions or a Pareto front to capture the trade-offs between objectives and allow the practitioner to choose the optimal compromise for their use case [1]. A Pareto front is a set of solutions to objective functions that do not dominate other solutions, mathematically defined as [1]:

$$\{f_i(a) \mid 1 \le i \le m,\ f_i(a) > f_i(b)\}$$

2.3 NAS and Search Algorithms

There are three parts to neural architecture search: (1) the *search space* that defines the structures to be designed and optimized, (2) the *search strategy* which defines the approach to be used to explore the search space and (3) the *performance estimation* that evaluates the performance of the structure. The three most common search strategies are reinforcement learning (RL), Bayesian optimization and evolutionary algorithms (EAs). RL-based search strategies are not ideal as they incur high computational cost and can take hours or days to complete [3]. Bayesian optimization typically is applied in low-dimensional problems, but can also work for NAS algorithms. Bayesian optimization, however, can struggle with the topographical structures of a neural network at times [3]. Evolutionary algorithms are inspired by biological evolution. EAs evolve the population in a common framework: mating selection, offspring generation, fitness evaluation and environmental selection [10].

3 Methodology

3.1 Search Space

The search space for each dataset varied according to dataset complexity and class size. Table 1 captures the variance in parameters for the DNN architectures for each dataset. Regardless of the dataset, each DNN is fully-connected and uses a rectified linear (ReLu) activation with linear transformations for each layer.

Dataset	Input Size	Output Size	Hidden Layers	Hidden Sizes	Batch Size
MNIST	784	10	3-20	10-200	128
CIFAR-10	3072	10	3-15	10-128	128

Table 1: Differences in parameters for the two datasets used to create the search space.

3.1.1 Datasets

Two datasets were used to create the search space: MNIST and CIFAR-10. Convolutional neural networks are known to outperform standard neural networks in imaging classification tasks, however, I provide two reasons for using MNIST and CIFAR-10 for this research: (1) A DNN model possesses the ability to surpass a CNN model in performance, simplicity, and computation time [14]. Although image-based CNN's may hold a higher accuracy, we save on computational efficiency and interpretability by using a DNN. (2) Several research papers and benchmarks use deep CNNs on MNIST and CIFAR-10 as baseline models [1, 2, 4, 5, 6, 12]. Applying these datasets on DNNs allows for direct comparisons with existing literature, providing context for the performance and interpretability of the proposed models.

3.1.2 Precomputed Objectives

To reduce the computational cost of the NAS, datasets containing trained and evaluated DNNs were created to speed up training time while exploring the search space. On each dataset (MNIST and CIFAR-10), 250 randomly generated architectures with the varying parameters shown in Table 1 were trained, evaluated and stored as data for rapid evaluation. This is the only time the architectures are trained and their objectives are evaluated, during the initialization of the search space. A decision tree regression model was employed to predict the objective values during the execution of the search strategy. The surrogate model uses five inputs to make a prediction: number of hidden layers, minimum and maximum hidden size, mean hidden size, and variance in hidden sizes. By using these precomputed objectives, hundreds of architectures can be evaluated in a matter of seconds and the computational complexity of the NAS is reduced. This method is similar to that of NAS-Bench-201, except there are far fewer architectures in the dataset and the surrogate model is not the same [13].

3.2 Search Strategy

The search space is explored using the Non-dominated Sorting Genetic Algorithm (NSGA-II), a computationally efficient evolutionary algorithm, optimizing three objectives: accuracy, introspectability and FLOPs (floating point operations) [11]. The algorithm samples a population of *N* architectures from the search space and evaluates their fitness (objectives) for *M* generations. These architectures are then used to generate new populations through a repetitive process of binary tournament, crossover and mutation. The crossover operator performs one-point and two-point crossover depending on the crossover probability, and the mutation operator either removes a hidden layer or modifies its size based on the mutation probability. Performing crossover and mutation accelerates the convergence rate of evolutionary NAS [10]. The algorithm works to maximize accuracy and introspectability and minimize the FLOPs, creating conflicting objectives [3] and caps the population size to *N* similar to method performed by [6]. Over time, this process improves and optimizes the objective values, creating ideal ANNs.

3.3 Performance Estimation

Each architecture aims to optimize three objectives: accuracy, introspectability and FLOPs. FLOPs calculate the computational complexity of deep learning models by counting the number of basic arithmetic operations that occur during the forward pass of the model. The accuracy metric is computed as the classification accuracy of each model. Introspectability metric was implemented from scratch as defined mathematically in (2.1) [1].

4 Experiments and Results

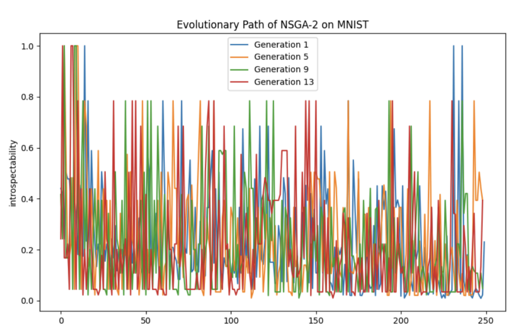
I evaluate EcoNAS on the two datasets – MNIST and CIFAR-10 – and conduct analyses to understand the evolution process of EcoNAS, examine trade-offs in introspectability and FLOPs in DNN structures, and compare results of EcoNAS to other benchmarks.

4.1 Evolutionary Paths

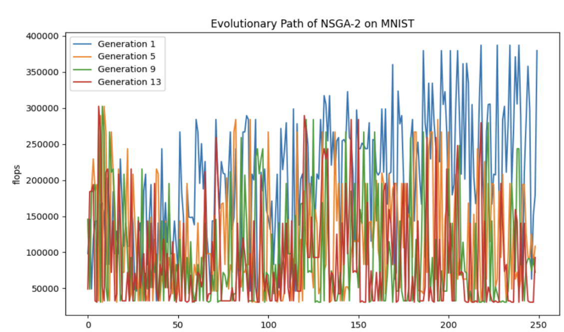
Dataset	Generations	Mean Accuracy	Mean Intros.	Mean FLOPs
MNIST	8 12 15	96.2% 96.8% 96.8%	0.327 0.313 0.289	116,239 105,764 86,148
CIFAR-10	8 12 15	43.7% 45.6% 47.3%	0.565 0.569 0.581	185,901 141,100 132,058

Table 2: Experimental results of the benchmarks for each dataset, listing mean accuracy, introspectability and FLOPs. The number of generations was kept the same for each dataset.

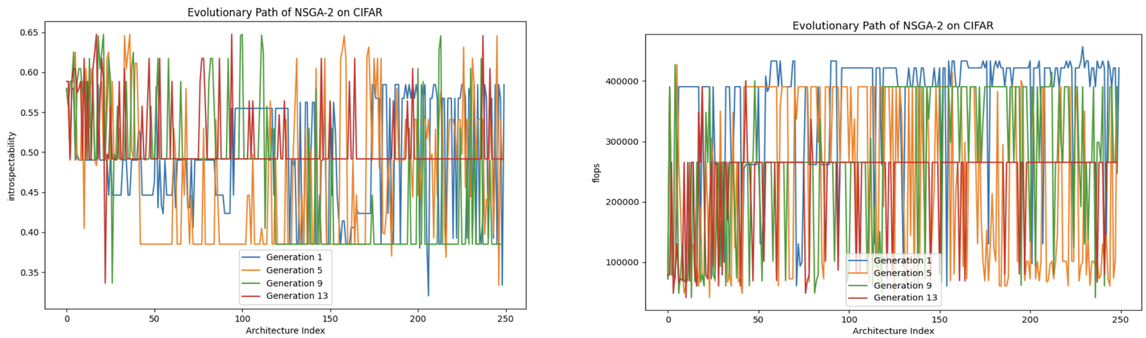
EcoNAS executed the NSGA-II algorithm three times on a different number of generations, with multi-objective optimization. The algorithm was run with a population of 250 randomly generated architectures from the MNIST and CIFAR-10 search spaces. Each randomly generated architecture in the population was created using the parameters from Table 1 with respect to their datasets. Crossover rates and mutation rates were fixed to 0.5 (50%).



(a) Evolutionary Path of Intros. MNIST



(b) Evolutionary Path of FLOPs MNIST



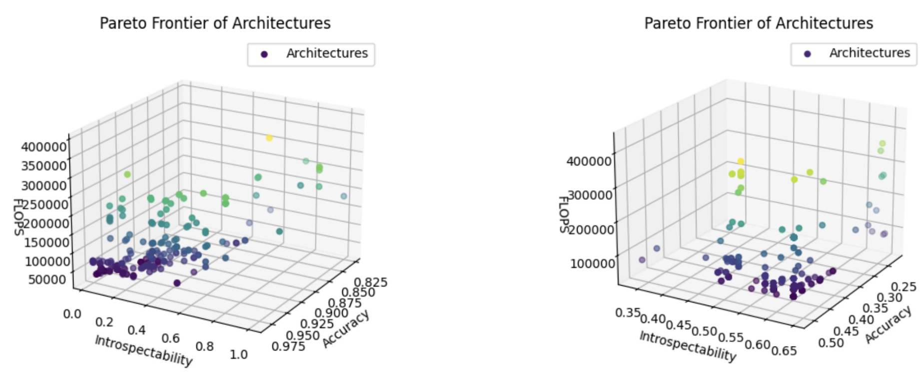
(c) Evolutionary Path of Intros. CIFAR-10 (d) Evolutionary Path of FLOPs. CIFAR-10

Figure 1: Evolutionary Paths of Objectives

At the end of each generation, the evolutionary path of the three objectives (accuracy, introspectability and FLOPs) were observed. Figures 1a-d demonstrate the evolutionary path of introspectability and FLOPs for each dataset for 15 generations. From Figure 1b, the algorithm succeeds in lowering the FLOPs over time while preserving the diversity in computational complexity. The performance of multi-objective optimization is well captured in the evolutionary path of the architectures in the CIFAR-10 dataset. On each increasing generation, the introspectability increased while the number of FLOPs decreased similar to the data shown in Table 2. The evolutionary path and convergence of values are crucial indicators of the algorithm's success in finding a set of diverse and Pareto-optimal solutions. The convergence of values does not lead to premature convergence to specific introspectability or FLOPs value but rather maintain a well-distributed set of solutions [11].

4.2 Analyzing Trade-offs

Both datasets captured a rather linear relationship between introspectability, FLOPs and accuracy. The MNIST dataset demonstrated a negative linear relationship, as shown in Figure 2a. As the accuracy increased, the mean introspectability and FLOPs decreased, indicating DNNs trained on the MNIST dataset lack interpretability, while being computational efficient and high-performing. XNAS has reported similar introspectability scores on the MNIST dataset, yet yielded slightly higher model performance [1].



(a) 3D plot of the architectures on MNIST (b) 3D plot of the architectures on CIFAR-10

Figure 2: Three dimensional graphs showing the plots of architectures and their objective values. The darker the points shows a higher concentration of architectures that had similar objective values.

However, the CIFAR-10 dataset responded in a different way: as accuracy increased, introspectability increased while FLOPs decreased (Figure 2b). These results challenge the reports of other benchmarks regarding the tradeoffs of computational complexity and accuracy [3, 8]. Moreover, the introspectability scores found by NAS are higher than those found by XNAS, and indicated overall better model performance on the CIFAR-10 dataset.

5 Discussion

As observed in the Table 2 and the above figures, there exists some tradeoff between the three metrics, accuracy, introspectability and FLOPs. This trade-off is more pronounced as the classification task grows more complex, as demonstrated in the MNIST and CIFAR-10 datasets. However, the CIFAR-10 dataset reveals it is possible to discover computationally efficient and interpretable architectures, although overall performance is quite low. The low accuracy presented by the CIFAR-10 can be explained by the use of an inefficient model (using DNN over CNN). Through this however, EcoNAS has not only challenged previous research supporting a trade-off in accuracy and interpretability, but research supporting a trade-off in computational complexity and accuracy.

6 Future Work

As the NAS-Bench-201 generates over 15,000 architectures, this work would benefit from the expansion of the precomputed objectives datasets and possibly generate architectures of higher complexity to capture extreme cases that may occur during the execution of the search strategy. Training the DNN architectures on other datasets may also yield comparable results to or challenge the results of other benchmarks, and could verify the results shown with the CIFAR-10 dataset. Reasonable datasets to experiment on may include CIFAR-100 and ImageNet [1, 3, 5].

7 Conclusion

I propose a new neural architecture benchmark, EcoNAS, which uses evolutionary multi-objective optimization approaches to create interpretable and computational efficient fee forward, fully-connected deep neural networks. Training the architectures on two image datasets, EcoNAS demonstrates the ability to improve accuracy, interpretability and computational efficiency without the presence of trade-offs. The experiments conducted also indicated convergence of each objective during the execution of the evolutionary algorithm NSGA-II. Once trained on additional image datasets, EcoNAS can provide researchers with high-performing, computationally efficient, interpretable deep learning structures.

References

[1] Zachariah Carmicheal, Tim Moon, Sam Ade Jacobs. *Learning Interpretable Models Through Multi-Objective NAS*. July 2023. https://arxiv.org/pdf/2112.08645.pdf

[2] Dong Dong, Hongxu Jiang, Xuekai Wei, Yanfei Song, Xu Zhuang, Jason Wang. *ETNAS: An energy consumption task-driven neural architecture search*. February 2023. https://doi.org/10.1016/j.suscom.2023.100926

[3] Shiqing Liu, Haoyu Zhang, Yaochu Jin. *Survey on computationally efficient NAS*. December 2022. https://doi.org/10.1016/j.jai.2022.100002

[4] Pedram Bakhtiarifard, Christian Igel, Raghavendra Selvan. *EC-NAS: Energy Consumption Aware Tabular Benchmarks for Neural Architecture Search*. May 2023. https://arxiv.org/pdf/2210.06015.pdf

[5] Chengyue Gong, Zixuan Jiang, Dilin Wang, Yibo Lin, Qiang Liu, and David Z. Pan. *Mixed Precision Neural Architecture Search for Energy Efficient Deep Learning*. 2019. https://ieeexplore.ieee.org/document/8942147

[6] Yiyang Zhao, Tian Guo. *Carbon-Efficient Neural Architecture Search*. July 2023. https://arxiv.org/pdf/2307.04131.pdf

[7] Matthias Feurer, Frank Hutter. *Hyperparameter Optimization*. 2019. https://link.springer.com/content/pdf/10.1007/978-3-030-05318-5_1.pdf

[8] Daniel T. Speckhard, Karolis Misiunas, Sagi Perel, Tenghui Zhu, Simon Carlile and Malcolm Slaney. *Neural Architecture Search for Energy Efficient Always-on Audio Machine Learning*. June 2023. https://arxiv.org/pdf/2202.05397v2.pdf

[9] Weishen Pan, Changshui Zhang. *The Definitions of Interpretability and Learning of Interpretable Models*. May 2021. https://arxiv.org/pdf/2105.14171.pdf

[10] Cheng he, Hao Tan, Shihua Huang, Ran Cheng. *Efficient evolutionary neural architecture search by modular inheritable crossover*. May 2021. https://www.sciencedirect.com/science/article/pii/S2210650221000559

[11] Kalyanmoy Deb , Amrit Pratap, Sameer Agarwal, T. Meyarivan. *A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II*. April 2002. https://ieeexplore.ieee.org/document/996017

[12] Zhichao Lu, Ran Cheng, Yaochu Jin, Kay Chen Tan, Kalyanmoy Deb. *Neural Architecture Search as Multiobjective Optimization Benchmarks: Problem Formulation and Performance Assessment*. April 2023. https://arxiv.org/pdf/2208.04321.pdf

[13] Xuanyi Dong, Yi Yang. *NAS-Bench-201: Extending the Scope of Reproducible Neural Architecture Search*. January 2020. https://arxiv.org/abs/2001.00326

[14] Farnaz Nazari, Wei Yan. *Convolutional versus Dense Neural Networks: Comparing the Two Neural Networks' Performance in Predicting Building Operational Energy Use Based on the Building Shape*. August 2021. https://arxiv.org/pdf/2108.12929.pdf