Comparing different search strategies on NAS-Bench-101

Supervisor: Prof. Dr. Michael Granitzer, Advisor: Julian Stier

Faculty of Computer Science and Mathematics, University of Passau

MUNIR, MUHAMMAD ARSAL

---

1 MOTIVATION

In Neural Architecture Search (NAS), the main idea is to minimize the human interference as much as possible to find better architectures from the given search space and to motivate the use of architecture search algorithms to discover them automatically. Because the design of network architecture is very significant regarding the final performance, therefore, nowadays researchers are working on different strategies to find out more suitable architecture to get optimal results. Similarly, in NAS-Bench-101, they have used random search and Bayesian optimization as their search strategies. But it seems like that using CMA-ES can produce better results as it is way better approach to achieve global maxima. And this evolution strategy is not explored much yet.

2 INTRODUCTION

In past few years, NAS has already made a major impact in many applications regarding neural networks. So now researchers are working on its improvement in different aspects. Especially, NAS is extremely slow in computing. It takes around few months which is not feasible for many researchers. Another problem is that everyone is working in this field in their own way, so their methods are unique and cannot be comparable, making it is really difficult to improve each other's methodologies. [7]. There are a lot of algorithms that are performing the search operation under the different search space. After getting these architectures, training has to be done using different approaches e.g., hyper-parameters, data augmentation, regularization. Each algorithm has its own approach therefore, it leads towards the comparability problem among each other. For tackling this problem, NAS-Bench-101 has already shown success to overcome this problem. [1]. Because, according to Chris Ying et al. [7] "NAS-Bench-101 exhaustively evaluates a search space, it permits, for the first time, a comprehensive analysis of a NAS search space as a whole". Additionally, Figure 1 shows a NAS framework.

Evolution strategies (ES) are the stochastic search algorithms, which use mutation, recombination, and selection applied to any population that consists of candidate solutions and iteratively evolve itself to get the better solutions and eventually evolve itself towards the optimal solution. There are many evolution strategies but in our work, we will discuss more about Covariance Matrix Adaptation Evolution Strategy (CMA-ES) and its pros and cons.

CMA-ES is a very useful search strategy using the covariance matrix to grow and shrink through the search space. CMA-ES is better to achieve global maxima within the given search space as compared to simple evolution strategy because in CMA-ES standard deviation (SD) is not fixed, otherwise, we will be stuck in local optima by using simple evolution strategy as SD is fixed. So it's the best evolution strategy if we have a reasonable amount of parameters [2]. It is one example of a usable optimization strategy for NAS (and many other search problems). The NAS is often used equivalent with an optimization problem. Thus, CMA-ES is only one of many possible strategies to search through the complex search

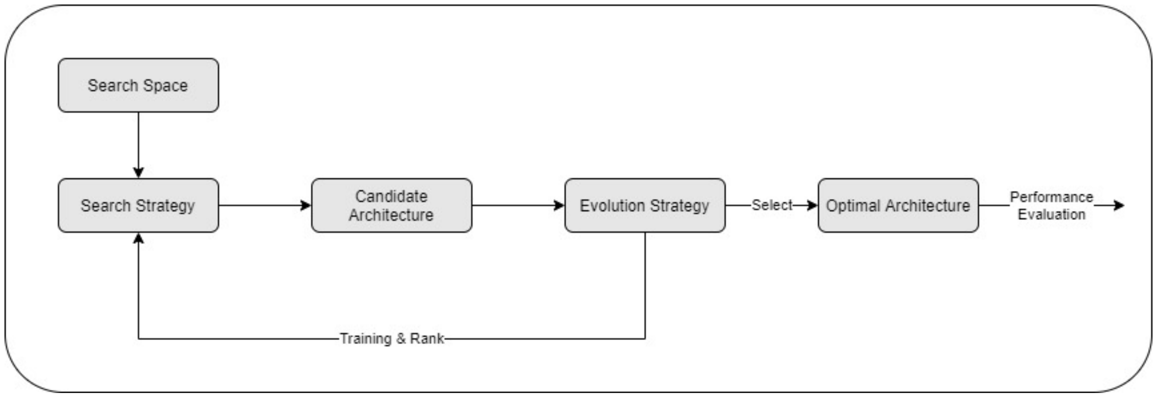

Fig. 1. The general framework of Neural Architecture Search (NAS). Generally, NAS starts with the predefined set of operations and from that search space it obtains a large number of network architectures. After getting these architectures set, candidate architecture is trained and ranked. Then obtaining new candidate network architectures after adjusting the search strategy by getting that ranking information. When the search is finished, the best network architecture is further checked for the final performance evaluation [5]

.

space. Stochastic Gradient Descent which is very popular for problems with differentiable search spaces and smooth spaces. It is actually used to optimize the training process of neural networks (which falls into "performance estimation" in the context of Neural Architecture Search, as you estimate the performance of architecture by training it on a dataset and the resulting score is an estimation of how good the particular architecture performs).

3 RELATED WORK

In April 2016, Ilya el at. from the University of Freiburg, published a paper about hyper-parameter optimization on Deep Neural Network (DNN) using CMA-ES. In which, they are claiming that “CMA-ES has some useful invariance properties and is friendly to parallel evaluations of solutions” [3]. They verified their approach on the MNIST dataset with 30 GPU in parallel.

In May 2019, a group of researchers published a paper claiming the first architecture dataset and benchmark for Neural Architecture Search, NAS-Bench-101 [7]. In which, they evaluated the whole NAS search space by comprehensive analysis as a whole. They explained different methodologies regarding the architecture search thoroughly. They trained and evaluated a large number of Convolutional Neural Networks (CNNs) on the CIFAR10 dataset. For architecture search, they have used different search strategies such as random search and Bayesian optimization.

In February 2018, Esteban Real el at. published a paper [4] regarding an evolution strategy to search image classification architectures from the given search space. They used a modified tournament selection evolution algorithm with an age property which works in favor of younger genotypes. They tested their approach on AmoebaNet-A and achieved comparable accuracy to current ImageNet models with complex architecture search methods. Additionally, NAS-Bench-201 [2] used this similar approach on CIFAR-10, CIFAR-100, and ImageNet datasets and achieved test accuracy of 93.92, 71.84, and 45.54 respectively.

In July 2020, Colin White el at. published a paper regarding the encodings for Neural Architecture Search. They defined different approaches of architecture encodings, such as adjacency matrix encodings and path-based encodings including a characterization of the scalability of each encoding. They run NAS algorithms on architectures encoded using these encoding schemes. After these experiments, these encodings can be differentiated from each other and we can have the best encoding depending on the architectures and algorithms. Both the encodings have their own pros and cons. In the adjacency matrix encoding scheme, a single architecture can yield multiple adjacency matrices, in contrast, in path-based encoding, multiple architectures can result in the same encoded path. Furthermore, they explained that NAS encodings are quite important because they can directly impact on the overall performance. [6]

4 RESEARCH GOALS

The research idea of my master thesis is to investigate different search strategies on NAS-Bench-101, especially the effects of CMA-ES. The main goal would be to give reproducible experiments for a CMA-ES and a random search and investigate on possible improvements of CMA-ES. I plan to achieve this goal by answering following research questions:

1. Is it possible to transform the NAS-Bench-101 search space into a continuous search space to apply e.g. CMA-ES as a search strategy?
2. How do Random Search, an Evolutionary Search, and possibly a genetic search compare on NAS-Bench-101?
3. How does CMA-ES perform on a possible continuous NAS-Bench-101 search space?
4. Can we exploit properties of the search space to estimate performances of NAS-Bench-101 without training it?

I will answer these questions by conducting several experiments on different datasets to make sure the approach I am following is generalized.

5 EXPERIMENTS

To investigate different search strategies on NAS-Bench-101, experiments, to begin with, are abstractly described as follows:

1. Apply Random search on the search space of NAS-Bench-101.
   - Train and test it on the dataset e.g CIFAR-10. We will use this result in comparison with other search strategies.
2. Change the representation of the architectures in the search space:
   - Search space of NAS-Bench-101 is not continuous. Therefore, we may not run CMA-ES directly on it. For this purpose, representation of the architectures has to be changed.
   - Encoding scheme can be used to encode architectures in the search space. For this purpose, adjacency matrix encoding can be used.
   - After that, we will get the vector representations and then evolution strategy will be implemented on it.
3. Compare the results of Random Search and CMA-ES on NAS-Bench-101:
   - Encode the graphs and convert them into vectors to run evolution strategy on it
   - Run CMA-ES on encoded architectures (Graphs), train and test it on CIFAR-10 and get the results in real numbers. Then evaluate it using performance metrics.
4. Find the relevant properties that have prominent impact on computing performance estimation of NAS-Bench-101,
   - Get the training data of different neural networks on CIFAR-10 dataset.

- Evaluate using performance metrics on respective architectures.
- Using a search strategy (e.g. Grid search), to identify the important properties that are impacting the performance.

6 EVALUATION

F1-Score is the weighted average of Precision and Recall. Therefore, this score considers both false positives and false negatives into account. Mostly accuracy is used when the class distribution is very much similar while F1-Score is a better metric when the distribution is imbalanced. In most cases in real-life problems, imbalanced class distribution exists. Plan to evaluate my experiments' results by calculating their F1-scores. For instance, we have to evaluate different search strategies such as Random Search on NAS-Bench-101 using CIFAR-10 dataset, we would know the result after sampling thousands of architectures from the search space. For evolution strategies like CMA-ES, it may require a continuous search space that contains vectors. So we have to train and test these vectors on CIFAR-10 and get the F-1 scores for each search strategy.

7 TIMELINE

The whole work is divided into four sub tasks i.e. research, implementation, evaluation, and documentation. The following Gantt chart depict the timeline that will be followed to complete the master thesis.

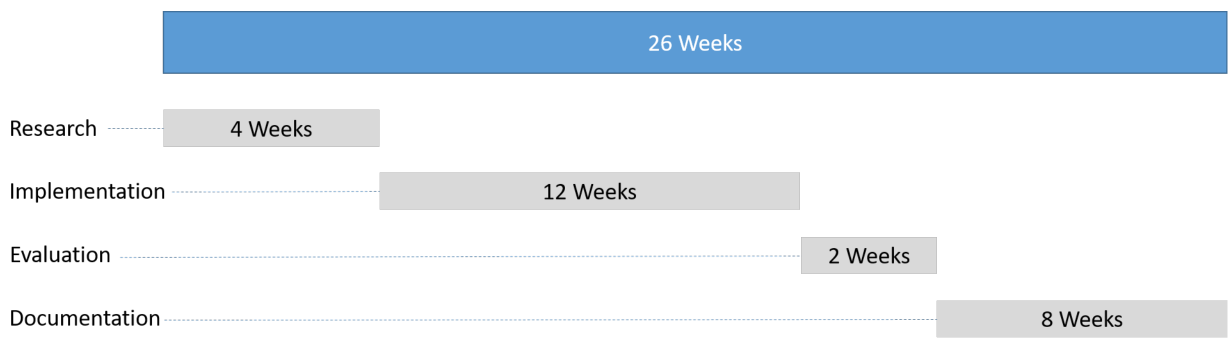

Fig. 2. Timeline of the thesis (26 weeks in total)

REFERENCES

[1] Xuanyi Dong and Yi Yang. *NAS-Bench-201: Extending the Scope of Reproducible Neural Architecture Search*. 2020. arXiv: 2001.00326 [cs.CV].

[2] Nikolaus Hansen and Andreas Ostermeier. "Completely Derandomized Self-Adaptation in Evolution Strategies". In: *Evol. Comput.* 9.2 (June 2001), 159–195. ISSN: 1063-6560. DOI: 10.1162/106365601750190398. URL: https://doi.org/10.1162/106365601750190398.

[3] Ilya Loshchilov and Frank Hutter. *CMA-ES for Hyperparameter Optimization of Deep Neural Networks*. 2016. arXiv: 1604.07269 [cs.NE].

[4] Esteban Real et al. *Regularized Evolution for Image Classifier Architecture Search*. 2018. arXiv: 1802.01548 [cs.NE].

[5] Pengzhen Ren et al. *A Comprehensive Survey of Neural Architecture Search: Challenges and Solutions*. 2020. arXiv: 2006.02903 `[cs.LG]`.

[6] Colin White et al. *A Study on Encodings for Neural Architecture Search*. 2020. arXiv: 2007.04965 `[cs.LG]`.

[7] Chris Ying et al. *NAS-Bench-101: Towards Reproducible Neural Architecture Search*. 2019. arXiv: 1902.09635 `[cs.LG]`.