An improved multi-objective genetic algorithm for the neural architecture search problem

Eloy Bedia-García[1] and Enrique Domínguez[1]

Dept. of Computer Science
University of Malaga
eloybg97@uma.es, enriqued@lcc.uma.es

1 Introduction

In recent years, there is a great interest in automating the process of searching for neural network topology. This problem is called Neural Architecture Search (NAS), which can be seen as a 3-gear mechanism: the search space, the error estimation and the search strategy.

The search space defines what kind of neural architectures can be reached, and it can be divided into three types:

- A search space restricted to sequential models. These models consist of consecutive layers $(L_0.L_1, ..., L_n)$, where layer L_i receives as input the output of layer L_{i-1}
- A search space with more complex architectures where the input of the i-th layer is defined by the function $g_i(L_0, ..., L_{i-1})$. Examples of this search space are the Residual Networks [4] or the DenseNets [5].
- The search space of block-based architectures [12], [11]. These are based on a design pattern that consists of the consecutive repetition of a set of layers called blocks.

To guide the selected strategy throughout the search space, we need a metric to help us. The simplest way is to evaluate the error obtained in the validation set, however, due to the long computation times required, alternative methods are being searched for, such as: reducing the training set [12], reducing the number of epochs [10, 12], using less filters [7, 12] or using lower resolution images [2].

Different kind of search strategies have been proposed in the literature mainly based on reinforcement learning with a reinforce policy [9], a proximal policy optimization [12] or Q-learning [1]. Other strategies based on evolutionary algorithms have been also proposed with different types of genetic operators, such as using selection by tournament [7, 8], eliminating the worst individual in each generation [8], eliminating the oldest individual [7], or using Lamarkian inheritance [3] for the offspring generation.

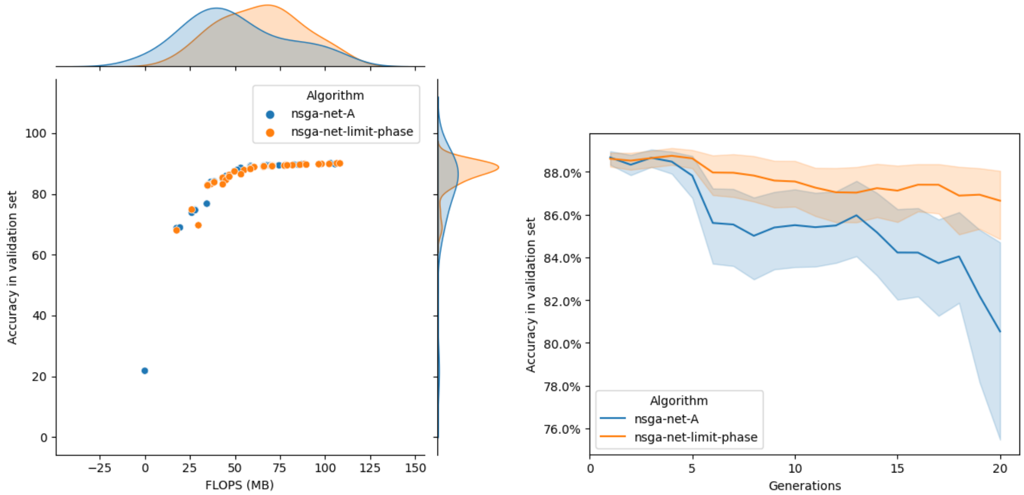
In this paper, we propose an improved version of the NSGA-Net algorithm [6], which is a multi-objective genetic algorithm for the NAS problem. One of the drawbacks is the limited diversity that can be generated by the original crossover operator, which generates only one offspring keeping the common genomes, and leaving the rest randomly. In order to avoid this limitation, we proposed a new 2-point crossover restricting the possible cutoff points only to the block limits.

The rest of the paper is organized as follows. In the next section, some preliminary results are shown on the well-known CIFAR-10 dataset. Finally, some remarks and future works are presented in the conclusions section.

2 Experiments

In this section, we present the empirical results to show the efficacy of the proposed operators for the NSGA-Net algorithm to automate the NAS process on the CIFAR-10 benchmark. Two objectives were considered to guide our NSGA-Net based algorithm: the classification error on the validation set and the computational complexity measured as the number of floating point operations (FLOPs) needed to execute the forward pass of the neural network. The CIFAR-10 dataset was considered for the classification task, splitting the original training set into our training (80%) and validation set (20%) for the neural architecture search. The original testing set was only used to obtain the test accuracy of the final models.

Regarding to the genetic parameters, the population was randomly initialized with size fixed to 40 during the 20 generations for exploration and 10 generations for exploitation. The coding of the genomes consists in a sequence of 3 blocks, each containing a maximum of 4 nodes. The training of the neural networks is carried out for 25 epochs, with a learning rate of 0.025 descending according to the cosine annealing scheme, and a batch size of 128.



(a) Objectives space distribution (b) Evolution of the population accuracy

Fig. 1: NSGA-Net vs LimitPhase on CIFAR-10

Figure 1 shows the comparative result between the original NSGA-Net and our proposed algorithm on the CIFAR-10 dataset. The distribution of the solutions provided by both algorithms is shown in figure 1a, where our proposal (NSGA-LimitPhase) presents a higher accuracy concentration in values around 90%, although with greater complexity (FLOPs). This information is also reflected in figure 1b, where the evolution of the accuracy of the population models of NSGA-LimitPhase presents a better performance. Additionally, in this figure we can see the deterioration of the NSGA-Net population throughout the generations, which it is accentuated at the end. However, our proposal presents accuracies more stable at high values with a smooth downward trend at the end.

3 Conclusions

In this work, we have proposed the use of a new crossover to improve the NSGA-Net algorithm for the NAS problem. Experimental results show a better accuracy distribution in the objectives space, and a better accuracy evolution on the CIFAR-10 dataset. These results are very promising for the treatment of other datasets/benchmarks in order to design a good algorithm for solving the NAS problem. Further works include both the application of the proposed algorithm to other datasets and the comparison to other state-of-the-art algorithms for the NAS problem.

Acknowledgement

The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga. They also gratefully acknowledge the support of NVIDIA Corporation with the donation of a RTX A6000 GPU with 48Gb.

References

1. Baker, B., Gupta, O., Naik, N., Raskar, R.: Designing neural network architectures using reinforcement learning
2. Chrabaszcz, P., Loshchilov, I., Hutter, F.: A downsampled variant of ImageNet as an alternative to the CIFAR datasets https://www.semanticscholar.org/paper/A-Downsampled-Variant-of-ImageNet-as-an-Alternative-Chrabaszcz-Loshchilov/e644a409b4a4c6eaedffe27efbc5c76280b34c61
3. Elsken, T., Metzen, J.H., Hutter, F.: Efficient multi-objective neural architecture search via lamarckian evolution. https://doi.org/10.48550/arXiv.1804.09081, http://arxiv.org/abs/1804.09081
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778. https://doi.org/10.1109/CVPR.2016.90, ISSN: 1063-6919
5. Huang, G., Liu, Z., Weinberger, K.: Densely connected convolutional networks p. 12
6. Lu, Z., Whalen, I., Boddeti, V., Dhebar, Y., Deb, K., Goodman, E., Banzhaf, W.: Nsga-net: Neural architecture search using multi-objective genetic algorithm. In: Proceedings of the Genetic and Evolutionary Computation Conference. p. 419–427. GECCO '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3321707.3321729, https://doi.org/10.1145/3321707.3321729
7. Real, E., Aggarwal, A., Huang, Y., Le, Q.V.: Aging evolution for image classifier architecture search. https://www.semanticscholar.org/paper/Aging-Evolution-for-Image-Classifier-Architecture-Real-Aggarwal/7bac3d11824fabe0dc3ac2fee9bfb667e82fba9c
8. Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y.L., Tan, J., Le, Q., Kurakin, A.: Large-scale evolution of image classifiers. https://doi.org/10.48550/arXiv.1703.01041, http://arxiv.org/abs/1703.01041
9. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning **8**(3), 229–256. https://doi.org/10.1007/BF00992696, https://doi.org/10.1007/BF00992696
10. Zela, A., Klein, A., Falkner, S., Hutter, F.: Towards automated deep learning: Efficient joint neural architecture and hyperparameter search. https://doi.org/10.48550/arXiv.1807.06906, http://arxiv.org/abs/1807.06906
11. Zhong, Z., Yan, J., Wu, W., Shao, J., Liu, C.L.: Practical block-wise neural network architecture generation. pp. 2423–2432. https://doi.org/10.1109/CVPR.2018.00257
12. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.: Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8697–8710. https://doi.org/10.1109/CVPR.2018.00907