On the Exploitation of Neuroevolutionary Information

Unai Garciarena*
Department of Computer Science and Artificial Intelligence
University of the Basque Country
Donostia-San Sebastian, Spain
unai.garciarena@ehu.eus

Nuno Lourenço
CISUC, Department of Informatics Engineering
Coimbra, Portugal
naml@dei.uc.pt

Penousal Machado
CISUC, Department of Informatics Engineering
Coimbra, Portugal
machado@dei.uc.pt

Roberto Santana
Department of Computer Science and Artificial Intelligence
University of the Basque Country
Donostia-San Sebastian, Spain
roberto.santana@ehu.eus

Alexander Mendiburu
Department of Computer Architecture and Technology
University of the Basque Country
Donostia-San Sebastian, Spain
alexander.mendiburu@ehu.eus

ABSTRACT

The final outcome of neuroevolutionary processes commonly is the best structure found during the search, and a good amount of *residual information* from which valuable knowledge that can be extracted is usually omitted. We propose an approach that extracts this information from neuroevolutionary runs, and use it to build a Bayesian network-based metamodel that could positively impact future neural architecture searches. The metamodel is learned from the best found solutions in previous GAN structural searches and it is used to improve subsequent neuroevolutionary searches.

CCS CONCEPTS

• **Computing methodologies** → **Discrete space search**; **Neural networks**; **Unsupervised learning**;

KEYWORDS

Generative adversarial networks, evolutionary algorithms, probabilistic graphical models, neural architecture search

ACM Reference Format:

Unai Garciarena, Nuno Lourenço, Penousal Machado, Roberto Santana, and Alexander Mendiburu. 2021. On the Exploitation of Neuroevolutionary Information. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459507

1 INTRODUCTION

Deep neural networks (DNN) have been successfully applied to many tasks, such as generative modeling, with, for example, generative adversarial networks (GAN) [4]. As the accessibility to augmenting computational resources has broadened, increasingly complex DNN structures have been proposed in the last few years, which has caused a drop-off in the feasibility of designing these models by hand. This has caused a considerable rise on the interest on neural architecture search (NAS) methods, a research field that encompasses methods aiming at automatizing structural design.

This work studies how to extract information about the characteristics of the best GANs found during previous searches to improve future searches. For that, we employ two use cases of evolved GANs. MLP-based GANs for making Pareto set (PS) approximations [3], and CNN-based GANs for generating images, COEGAN [2]. We employ Bayesian networks (BN) [1] to model the characteristics of the best GANs, as they are interpretable, easy to sample, and are able to capture dependencies between GAN characteristics.

2 BACKGROUND

GANs are composed of two DNNs, a generator and a discriminator. While the latter aims at being able to discern real data observations from fake ones, the former pursues fooling the discriminator by generating data samples as similar to the real ones as possible. The final objective of the GAN is to learn a realistic generative model.

A generator, partially defined by its parameters θ_G, receives random noise and produces samples in the space of the original data $\mathbf{x}$: $G(\mathcal{N}(\mu,\sigma),\theta_g) \rightarrow \hat{\mathbf{x}}$. The discriminator, using its parameters θ_d, receives either $\mathbf{x}$or $\hat{\mathbf{x}}$ as input, and provides probability values of the inputs emanating from $\mathbf{x}$.

BNs are PGMs used to represent sets of variables and their (in)dependencies, using directed acyclic graphs (DAG). Each node of the DAG represents a variable, and the (non) existence of an arc between two nodes represents the (in)dependency between them.

The BN-based metamodel has been designed in two levels in order to simplify the modeling of dynamic DNN depths by the BNs. The first level, the *supermodel*, fixes the number of layers of the DNNs. In the second level, one BN is learned for each DNN depth.

In this work the ARACNE algorithm was used to learn the structure of the BN, as implemented in the bnlearn R library [8].

3 SEARCH INITIALIZATION

The NE runs in [3] consist of 30 runs (100 generations, 100 individuals/generation) for GAN structural search with the final goal

*Corresponding author.

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3459507

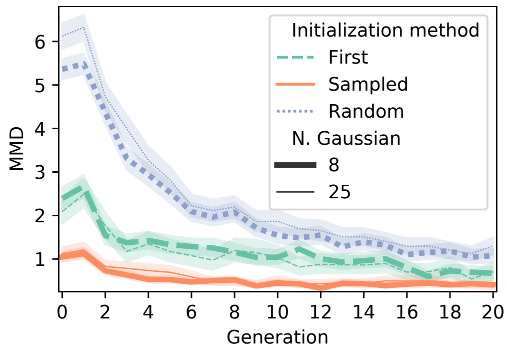

Figure 1: Best MMD value (y axis) corresponding to the samples generated by GANs at different generations (x axis) during NE. The solid lines represent the mean of the $20inds/gen \times 30runs = 60$ computed MMDs at each generation. The translucent bands show the 95% confidence interval.

of generating PS approximations of the functions from the suite defined in [6] (except for $F6$).

We define the *Best* set, which consists of the 5 best performing GAN structures of each run. Over 75% of the GANs in the *Best* are combinations of generators and discriminators of depth up to three and four layers respectively. For reducing the complexity of the metamodel, we limit the GANs to these $3 \times 4 = 12$ combinations. A metamodel is learned using the *Best* set and sampled 100 times, to conform the *Sampled* set.

We design a NE search for GANs (of 20 generations of 20 individuals each) for the 2D 8 and 25-Gaussian approximation problem [7]. 30 runs of three versions of the NE algorithm are run, each of them initialized using GANs randomly, or from the *Best*, or *Sampled* sets. Figure 1 shows the per generation evolution of the best GAN in terms of Maximum Mean Discrepancy (MMD) [5], the fitness function used to evaluate the quality of a GAN (the second objective of the bi-objective evolutionary process being the minimization of the elapsed time during training and sampling the GANs).

As can be seen in the figure, the evolutionary runs with the non-random initialization have a large advantage in the initial stages of the evolution. The GANs found in the runs initialized with the *Sampled* set clearly outperform the other two approaches.

4 SEARCH GUIDE

Firstly, COEGAN is run for the Fashion MNIST [9] database 20 times. Next, a *Best* set of GANs was created, and a metamodel was learned, as in Section 3. 30 different runs of two variants of a hill climbing (HC) algorithm are executed with 100 evaluations, looking for GAN structures which can accurately reproduce images similar to the digits available in the MNIST dataset. The first variant randomly generates a neighbor of the current solution and evaluates it. The second one randomly generates all the possible neighbors (in the COEGAN search space), and only evaluates the one to which the metamodel awards the larger probability (the one the metamodel considers to be the most promising one).

Figure 2 shows, for each step in the HC procedure (x axis), the best found FID value (y axis). The figure shows that, during the first

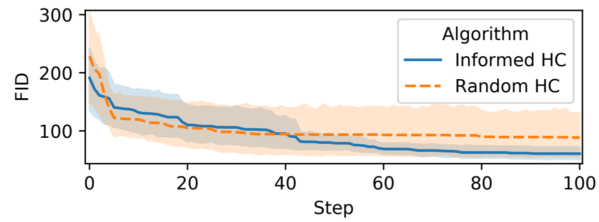

Figure 2: Evolution of the FID (in the y axis) of the best CNN-GAN structure found at each step (x axis).

40 steps of the search, both HC procedures show similar behaviours. In the second part of the search, only the guided greedy algorithm is capable of showing steady improvement.

5 CONCLUSIONS

The results reported in this work clearly show the advantages of using the metamodel for initializing or guiding NE procedures. This speaks for the versatility of the approach, as the same metamodel was able to perform both tasks efficiently.

ACKNOWLEDGMENTS

This work has received support from the TIN2016-78365-R (Spanish Ministry of Economy, Industry and Competitiveness), PID2019-104966GB-I00 (Spanish Ministry of Science and Innovation), IT-1244-19 (Basque Government), KK-2020/00049 (part of the project 3KIA, funded by the SPRI-Basque Government through Elkartek) programs, and by Portuguese national funds through the FCT - Foundation for Science and Technology, I.P., through the project CISUC - UID/CEC/00326/2020. Unai Garciarena holds apredoctoral grant (PIF16/238) by the University of the Basque Country.

REFERENCES

[1] E. Castillo, J. M. Gutierrez, and A. S. Hadi. 1997. *Expert Systems and Probabilistic Network Models*. Springer.

[2] Victor Costa, Nuno Lourenço, and Penousal Machado. 2019. Coevolution of Generative Adversarial Networks. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*. Springer, 473–487. https://doi.org/10.1007/978-3-030-16692-2_32

[3] Unai Garciarena, Alexander Mendiburu, and Roberto Santana. 2020. Analysis of the transferability and robustness of GANs evolved for Pareto set approximations. *Neural Networks* 132 (2020), 281–296. Publisher: Elsevier.

[4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. MIT Press, Cambridge, MA, USA, 2672–2680. https://doi.org/10.5555/2969033.2969125 event-place: Montreal, Canada.

[5] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A Kernel Two-Sample Test. *Journal of Machine Learning Research* 13, 25 (2012), 723–773. http://jmlr.org/papers/v13/gretton12a.html

[6] H. Li and Q. Zhang. 2008. Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation* 13, 2 (2008), 284–302. https://doi.org/10.1109/TEVC.2008.925798

[7] L Metz, B Poole, D Pfau, and J Sohl-Dickstein. 2016. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163* [cs,stat] (2016). https://arxiv.org/abs/1611.02163

[8] Marco Scutari and Jean-Baptiste Denis. 2014. *Bayesian Networks: With Examples in R.* CRC press.

[9] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. (Aug. 2017). arXiv: cs.LG/1708.07747.