Adversarial Deep Learning by Using Coevolutionary Computation

Jamal Toutouh[1,2], Una-May O'Reilly[1]
[1]*CSAIL, Massachusetts Institute of Technology* Cambridge, MA, USA
[2]*ITIS Software. Univeristy of Malaga* Malaga, Spain
jamal@uma.es, unamay@csail.mit.edu

I. Introduction

In recent years, machine learning with Generative Adversarial Networks (GANs) has been recognized as a powerful method for generative modeling. Generative modeling is the problem of estimating the underlying distribution of a set of samples. GANs accomplish this using unsupervised learning. They have also been extended to handle semi-supervised and fully supervised learning paradigms. GANs have been successfully applied to many domains. They can generate novel images (e.g., image colorization or super-resolution, photograph editing, and text-to-image translation), sound (e.g., voice translation and music generation), and video (e.g., video-to-video translation, deepfakes generation, and AI-assisted video calls), finding application in domains of multimedia information, engineering, science, design, art, and games.

GANs apply an adversarial paradigm. GAN training can be seen as a two-player game in which two neural networks compete with each other using an antagonistic loss function to train the parameters with gradient descent. This learning paradigm connects GANs to evolution because evolution also exhibits adversarial engagements and competitive co-evolution. In fact, co-evolutionary algorithms (COEAs) offer a means of solving convergence impasses often encountered in GAN training.

II. Outline

In this tutorial, we will explain:

- Main concepts of generative modeling and adversarial learning. GAN gradient-based training and the main pathologies that prevent ideal convergence. We will explain mode collapse, oscillation, and vanishing gradients.
- Co-evolutionary algorithms and how they can be applied to train GANs. Specifically, we will explain how algorithm enhancements address non-ideal convergence.
- We will draw upon the open-source Lipizzaner framework to demonstrate (url: http://lipizzaner.csail.mit.edu/). This framework is easy to use and extend. It sets up a spatial grid of communicating populations of GANs.
- Students will be given the opportunity to set up and use the Lipizzaner framework during the tutorial employing a jupyter notebook expressly developed for teaching purposes.

Main publications that define the background in the tutorial area are listed in the **References**.

III. Tutorial main details

Classification: Advanced.
Duration: 2 hours.
Target audience: 50 researchers and PhD students from the evolutionary computation and machine learning community.

IV. Websites

The main websites related to the tutorial are the following:

- Lipizzaner project: http://lipizzaner.csail.mit.edu/
- ALFA Research Group: https://alfagroup.csail.mit.edu/
- Websites of Lipizzaner project leaders Jamal Toutouh (http://jamal.es/) and Una-May O'Reilly (https://alfagroup.csail.mit.edu/unamay)

V. Presenters

Jamal Toutouh is a Associate Professor at the University of Málaga (Spain). Previously, he was a Marie Skłodowska Curie Postdoctoral Fellow at Massachusetts Institute of Technology (MIT) in the USA, at the MIT CSAIL Lab. He obtained his Ph.D. in Computer Engineering at the University of Malaga (Spain). His Ph.D Thesis was awarded the 2018 Best Spanish Ph.D. Thesis in Smart Cities. His current research explores the combination of Nature-inspired gradient-free and gradient-based methods to address Machine Learning. The main idea is to devise new algorithms to improve the efficiency and efficacy of the state-of-the-art methodology by mainly applying co-evolutionary approaches. Besides, he is working on the application of Machine Learning to address problems related to Smart Mobility, Smart Cities, and Climate Change. Web: http://jamal.es/

Prof. Una-May O'Reilly is the leader of the AnyScale Learning For All (ALFA) group at MIT CSAIL. ALFA focuses on evolutionary algorithms, machine learning, and frameworks for large-scale knowledge mining, prediction and analytics. The group has projects in cyber security using coevolutionary algorithms to explore adversarial dynamics in networks and malware detection. Una-May received the EvoStar Award for Outstanding Achievements in Evolutionary Computation in Europe in 2013. She is a Junior Fellow (elected before age 40) of the International Society of Genetic and Evolutionary Computation, which has evolved into ACM Sig-EVO. She now serves as Vice-Chair of ACM SigEVO. Web: https://alfagroup.csail.mit.edu/unamay

VI. Tutorial Activities and learning outcomes

The three main teaching goals of this tutorial are: *1.* Introduce the main concepts of GANs, *2.* Present the benefits of using COEAs to assist with train GANs, and *3.* Demonstrate how to use the Lipizzaner framework to obtain generative models. Thus, the content is grouped into three blocks:

1) **Introduction to GANs**: Overview of deep learning. Examples of GAN application. Describe the main (adversarial) concepts related to GANs including the min-max formulation. Introduce standard approaches to GAN training. Present different GAN variations such as the Conditional GANs and Cycle GANs. Introduce GAN training pathologies.
2) **Evolutionary computing meets GAN training**: Here, we show how COEAs are used to train GANs overcoming main pathologies. Thus, we review the main concepts of COEAs. We present some real examples from Nature and different optimization problems. After that, we introduce different coevolutionary approaches to address GAN training.
3) **Using Lipizzaner for robust and resilient GAN training**: The last part of our tutorial comprises a series of hands-on activities to better understand COEA GAN training. The main activity is to present and use Lipizzaner, a free open-source framework for training GANs by using a spatially distributed co-evolutionary algorithm. Therefore, after a quick installation, we run some examples, which take just a few cycles to train. As soon as the audience gets familiar with how to run such examples, we present how to tune and configure the methods by using the experimentation configuration files and how to extend the code to develop their own COEAs. To help the attendees self-assess their learning outcomes, we propose and cover a couple of exercises using state-of-the-art datasets (e.g., MNIST and CIFAR-10).

Attendees will, at the end of the tutorial, be able to: *a)* identify the intellectual intersection between GANs and coevolutionary algorithms, *b)* describe the design principles of a GAN and be familiar with simple code for one, and *c)* use Python code to run a demonstration framework.

VII. Tutorial previous history

Some earlier variations of this tutorial were presented in the following venues:

- Coevolutionary Computation for Adversarial Deep Learning during the **Genetic and Evolutionary Computation Conference (GECCO) 2024** (hybrid) with an audience of about 50 in-person and online assistants. July 2024.
- Coevolutionary Computation for Adversarial Deep Learning during the **Genetic and Evolutionary Computation Conference (GECCO) 2023** (hybrid) with an audience of about 50 in-person and online assistants. July 2023.
- Robust and resilient GAN training through spatially distributed coevolution at the **Centre for Informatics and Systems (University of Coimbra, Portugal)**, March 2022.
- Robust and Resilient Generative Adversarial Networks Training by using Distributed Coevolution, at the **School of Computer Science and Engineering (University of Malaga, Spain)**, November 2022.
- Adversarial Deep Learning by Using Distributed Coevolutionary Computation during the International Conference on **Parallel Problem Solving from Nature, PPSN XVII**, (in person) with an audience of 45 assistants. September 2022.
- Coevolutionary Computation for Adversarial Deep Learning during the **Genetic and Evolutionary Computation Conference (GECCO) 2022** (hybrid) with an audience of about 50 in-person and 34 online assistants. July 2022.
- Adversarial Deep Learning by Using Coevolutionary Computation during **the IEEE Congress on Evolutionary Computation (CEC) 2021** (virtual) with an audience of 100 assistants. July 2021.
- Coevolutionary Computation for Adversarial Deep Learning during the **Genetic and Evolutionary Computation Conference (GECCO) 2021** (virtual) with an audience of 100 assistants. July 2021.
- Lipizzaner: Distributed Coevolution for Resilient Generative Adversarial Networks Training, at the **Faculty of Engineering (Univ. de la Republica, Uruguay)** as a webinar with an audience of 30 assistants, April 2020.

References

[1] T. Schmiedlechner, I. Yong, A. Al-Dujaili, E. Hemberg, and U. O'Reilly, "Lipizzaner: A System That Scales Robust Generative Adversarial Network Training," in *32^{nd} Conference on Neural Information Processing Systems*, 2018.

[2] J. Toutouh, E. Hemberg, and U.-M. O'Reilly, "Spatial evolutionary generative adversarial networks," in *Genetic and Evolutionary Computation Conference*, 2019, pp. 472–480.

[3] ——, "Data dieting in gan training," in *Deep Neural Evolution*. Springer, 2020, pp. 379–400.

[4] ——, "Analyzing the components of distributed coevolutionary gan training," in *The Sixteenth International Conference on Parallel Problem Solving from Nature (PPSN XVI).*, 2020, p. 14.

[5] J. Toutouh and U.-M. O'Reilly, *Signal Propagation in a Gradient-Based and Evolutionary Learning System*. New York, NY, USA: Association for Computing Machinery, 2021, p. 377–385.

[6] E. Hemberg, J. Toutouh, A. Al-Dujaili, T. Schmiedlechner, and U.-M. O'Reilly, "Spatial coevolution for generative adversarial network training," *ACM Trans. Evol. Learn. Optim.*, vol. 1, no. 2, jul 2021.

[7] A. Al-Dujaili, T. Schmiedlechner, E. Hemberg, and U.-M. O'Reilly, "Towards distributed coevolutionary GANs," in *AAAI 2018 Fall Symposium*, 2018.

[8] J. Toutouh, E. Hemberg, and U.-M. O'Reilly, "Adversarial evolutionary learning with distributed spatial coevolution," in *Handbook of Evolutionary Machine Learning*. Springer, 2023, pp. 397–435.

[9] J. Toutouh, S. Nalluru, E. Hemberg, and U.-M. O'Reilly, "Semi-supervised generative adversarial networks with spatial coevolution for enhanced image generation and classification," *Applied Soft Computing*, vol. 148, p. 110890, 2023.

[10] J. Toutouh, S. Nalluru, E. Hemberg, and U.-M. O'Reilly, "Semi-supervised learning with coevolutionary generative adversarial networks," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2023, pp. 568–576.

[11] D. Flores, E. Hemberg, J. Toutouh, and U.-M. O'Reily, "Coevolutionary generative adversarial networks for medical image augumentation at scale," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2022, pp. 367–376.

[12] M. Hevia Fajardo, P. K. Lehre, J. Toutouh, E. Hemberg, and U.-M. O'Reilly, "Analysis of a pairwise dominance coevolutionary algorithm with spatial topology," in *Genetic Programming Theory and Practice XX*. Springer, 2024, pp. 19–44.

[13] P. K. Lehre, M. Hevia Fajardo, J. Toutouh, E. Hemberg, and U.-M. O'Reilly, "Analysis of a pairwise dominance coevolutionary algorithm and defendit," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2023, pp. 1027–1035.

[14] S. Nesmachnow, J. Toutouh, G. Ripa, A. Mautone, and A. Vidal, "Parallel-distributed implementation of the lipizzaner framework for multiobjective coevolutionary training of generative adversarial networks," in *Latin American High Performance Computing Conference*. Springer, 2023, pp. 97–112.

[15] G. Ripa, A. Mautone, A. Vidal, S. Nesmachnow, and J. Toutouh, "Multi-objective coevolutionary training of generative adversarial networks," in *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, 2023, pp. 319–322.

[16] M. A. Hevia Fajardo, E. Hemberg, J. Toutouh, U.-M. O'Reilly, and P. K. Lehre, "A self-adaptive coevolutionary algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2024, pp. 841–849.

[17] E. Hemberg, U.-M. O'Reilly, and J. Toutouh, "Cooperative coevolutionary spatial topologies for autoencoder training," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2024, pp. 331–339.