Neuroevolution of Augmenting Topologies

Paul Pauls

Advisor: Michael Adam

CONTENTS

Abstract—Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

I. INTRODUCTION

THIS shall be my introduction. And this shall be my citation [**?**]. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

II. NEUROEVOLUTION AND EVOLUTIONARY ALGORITHMS

Neuroevolution is a machine learning technique that applies evolutionary algorithms to construct artificial neural networks, taking inspiration from the evolution of biological nervous systems in nature. [**?**]

A evolutionary algorithm is a generic population-based and meta-heuristically optimized algorithmic solution to an applied problem. When breaking this down to simpler terms, does this mean first of all that an evolutionary algorithm (short form: **EA**) is simply a solution to an applied problem. This solution can take the form of a classical calculation algorithm or more complex forms like the aforementioned artificial neural network. As generic as its method of solving the problem is, is its application domain. Evolutionary Algorithms - as well as Neuroevolution by which they are employed - are highly general and allow for learning without explicit targets even if provided with only minimal feedback. [**?**]

Evolutionary Algorithms are population-based, meaning they not only handle a single solution to the problem they are applied to, but they have a multitude of solutions to this problem. Each of the solutions is called a *member* of its arbitrarily large population and each solution is arbitrarily similar to one another. Each member of the population is judged by a common *fitness function*, which numerically expresses the members quality of its solution to the applied problem. The higher the corresponding *fitness function* score of a member, the better is the solution to the applied problem. All members are judged by the same fitness function, which is the key hyperparameter that determines how well a problem has been solved.

The key aspect of evolutionary algorithms however lies in its meta-heuristic optimization method. This optimization method improves the members of the population in the sense that along the evolutionary process almost all members of the population score an increasingly higher fitness function evaluation score - meaning they get increasingly better at solving the problem they are applied to. This optimization method is employed after each *generation* of a population. A generation in the evolutionary process is completed once every member of the population has been applied to the problem and has been assigned a fitness score by the common fitness function. The population is then mixed up through *reproduction*, *mutation*, *recombination* and *selection*. These processes spread traits of high-performing members to low-performing members, preserve high-performing members while extinguishing low-performing members and introduce novel traits to already high-performing members to further explore the solution-space.

Applying the attributes of Evolutionary Algorithms to ar-

tificial Neural Networks does result in the machine learning technique Neuroevolution. Neuroevolution algorithms start out with an initial population of neural networks of varying complexity - meaning topology and weights. The exact nature of this initial population is not stipulated by the Neuroevolution technique and depends on the specific Neuroevolution algorithm. In each generation does each neural network member perform input inference and is subsequently judged by the fitness function. The best scoring neural networks spread their traits to lower scoring neural network. Fixed-Topology Neuroevolution algorithms solely adjust the weights of the lower scoring neural networks towards the higher scoring networks while Topology and Weight Evolving Artificial Neural Network algorithms (short form: **TWEANNs**) additionally also adjust the topology. They introduce new nodes and connections or delete nodes and connections in the low scoring neural networks depending on if they are present in the better scoring neural networks or introduce completely novel nodes and connections in order to explore new, possibly even better performing, solutions. Through this do neural networks developed through Neuroevolution algorithms often have arbitrary neural models and network structures and are able to progress in learning even without explicit targets and only sparse feedback.

While concluding this introduction to Neuroevolution can be said that Neuroevolution can be classified as part of the machine learning paradigm of *Reinforcement Learning*. Neuroevolution algorithms learn through ever-changing interaction with the environment instead of a predetermined set of input-output mapping or creative self-organization of input through probabilistic models. Neuroevolution algorithms learn from the consequence of their actions on the basis of their past experiences and also by new choices - quite accurately conforming with the definition of Reinforcement learning.

A. Landmark Research in Neuroevolution

III. NEUROEVOLUTION OF AUGMENTING TOPOLOGIES (NEAT)

A. ¡Section Introduction¿

B. Key Aspects of NEAT and Differences to Preceding Neuroevolution

C. Performance of NEAT

D. Variants and Advancements of NEAT

IV. APPLICATIONS OF NEAT

V. CONCLUSIONS

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit

blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

REFERENCES

[1] Yao - Evolving Artificial Neural Networks; 1999; http://avellano.fis.usal.es/~lalonso/compt_soft/articulos/yao99evolving.pdf

[2] Stanley, Miikkulainen - Efficient Evolution of Neural Network Topologies; 2002; http://nn.cs.utexas.edu/downloads/papers/stanley.cec02.pdf

[3] Stanley, Miikkulainen - Evolving Neural Networks through Augmented Topologies; 2002; http://nn.cs.utexas.edu/downloads/papers/stanley.ec02.pdf

[4] Geard, Wiles - Structure and Dynamics of a Gene Network Model Incorporating Small RNAs; Dec 2003; https://ieeexplore.ieee.org/document/1299575

[5] Stanley - Efficient Evolution of Neural Networks through Complexification; Aug 2004; http://nn.cs.utexas.edu/downloads/papers/stanley.phd04.pdf

[6] Reisinger, Miikkulainen - Acquiring Evolvability through Adaptive Representations; Jul 2007; http://nn.cs.utexas.edu/downloads/papers/reisinger.gecco07.pdf

[7] Mattiussi, Duerr, et al - Center of Mass Encoding: A self-adaptive representation with adjustable redundancy for real-valued parameters; Jul 2007; https://infoscience.epfl.ch/record/101405

[8] Floreano, Duerr, et al - Neuroevolution: From Architectures to Learning; Jan 2008; http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.182.1567

[9] Stanley, D'Ambrosio, et al - A Hypercube-Based Indirect Encoding for Evolving Large-Scale Neural Networks; 2009; http://axon.cs.byu.edu/~dan/778/papers/NeuroEvolution/stanley3**.pdf

[10] Risi, Stanley - Enhancing ES-HyperNEAT to Evolve More Complex Regular Neural Networks; Jul 2011; http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.365.4332

[11] Lehman, Stanley - Novelty Search and the Problem with Objectives; Oct 2011; https://www.cs.ucf.edu/eplex/papers/lehman_gptp11.pdf

[12] Woergoetter, Porr - Scholarpedia Article on 'Reinforcement Learning'; Sep 2012; http://www.scholarpedia.org/article/Reinforcement_learning

[13] Holland - Scholarpedia Article on 'Genetic Algorithms'; Oct 2012; http://www.scholarpedia.org/article/Genetic_algorithms

[14] Fogel, Fogel, et al - Scholarpedia Article on 'Evolutionary Programming'; Oct 2013; http://www.scholarpedia.org/article/Evolutionary_programming

[15] Lehman, Miikkulainen - Scholarpedia Article on 'Neuroevolution'; Oct 2013; http://www.scholarpedia.org/article/Neuroevolution

[16] Pascanu, Ganguli, et al - On the Saddle Point for Non-Convex Optimization; May 2014; https://www.researchgate.net/publication/262452520

[17] Kim, Minyoung, et al - Deep Clustered Convolutional Kernels; Mar 2015; https://arxiv.org/abs/1503.01824

[18] Fernando, Banarse, et al - Convolution by Evolution; Jun 2016; https://arxiv.org/abs/1606.02580

[19] Miikkulainen, Liang, et al - Evolving Deep Neural Networks; Mar 2017; https://arxiv.org/abs/1703.00548

[20] Xie, Yuille - Genetic CNN; Mar 2017; https://arxiv.org/abs/1703.01513

[21] Negrinho, Gordon - DeepArchitect: Automatically Designing and Training Deep Architectures; Apr 2017; https://arxiv.org/abs/1704.08792

[22] Real, Moore, et al - Large-scale Evolution of Image Classifiers; Jun 2017; https://arxiv.org/abs/1703.01041

[23] Stanley - Neuroevolution: A Different Kind of Deep Learning; Jul 2017; https://www.oreilly.com/ideas/neuroevolution-a-different-kind-of-deep-learning

[24] Brock, Lim, et al - SMASH: One-Shot Model Architecture Search through HyperNetworks; Aug 2017; https://arxiv.org/abs/1708.05344

[25] Salimans, Ho - Evolution Strategies as a Scalable Alternative to Reinforcement Learning; Sep 2017; https://arxiv.org/abs/1703.03864

[26] Jaderberg, Dalibard, et al - Population Based Training of Neural Networks; Nov 2017; https://arxiv.org/abs/1711.09846

[27] Zhang, Clune, et al - On the Relationship Between the OpenAI Evolution Strategy and Stochastic Gradient Descent; Dec 2017; https://arxiv.org/abs/1712.06564

[28] Stanley, Clune - Welcoming the Era of Deep Neuroevolution; Dec 2017; https://eng.uber.com/deep-neuroevolution/

[29] Liu, Simonyan, et al - Hierarchical Representation for Efficient Architecture Search; Feb 2018; https://arxiv.org/abs/1711.00436

[30] Such, Stanley, et al - Accelerating Deep Neuroevolution: Train Atari in Hours on a Single Personal Computer; Apr 2018; https://eng.uber.com/accelerated-neuroevolution/

[31] Such, Madhavan, et al - Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning; Apr 2018; https://arxiv.org/abs/1712.06567

[32] Zoph, Vasudevan, et al - Learning Transferable Architectures or Scalable Image Recognition; Apr 2018; https://arxiv.org/abs/1707.07012

[33] Lehman, Chen, et al - ES Is More Than Just a Traditional Finite-Difference Approximator; May 2018; https://arxiv.org/abs/1712.06568

[34] Zhong, Yan, et al - Practical Block-Wise Neural Network Architecture Generation; May 2018; https://arxiv.org/abs/1708.05552

[35] Rawal, Miikkulainen - From Nodes to Networks: Evolving Recurrent Neural Networks; Jun 2018; https://arxiv.org/abs/1803.04439

[36] Conti, Madhavan, et al - Improving Exploration in Evolution Strategies for Deep Reinforcement Learning via a Population of Novelty Seeking Agents; Oct 2018; https://arxiv.org/abs/1712.06560

[37] Real, Aggarwal, et al - Regularized Evolution for Image Classifier Architecture Search; Feb 2019; https://arxiv.org/abs/1802.01548

[38] Sun, Xue, et al - Evolving Deep Convolutional Neural Networks for Image Classification; Mar 2019; https://arxiv.org/abs/1710.10741