Speed's the Game

For the development of behaviorally diverse *Sonic the Hedgehog* agents

Alex Nazaruk, Southwestern University

Introduction

Last year's OpenAI Retro Contest in the game of Sonic the Hedgehog ended with no submitted agent beating all the test levels. We propose a three-component solution aimed at creating diverse play styles. Evolution combined with a behavioral novelty objective and deep reinforcement learning is used to create skilled Sonic the Hedgehog agents.

Sonic as a benchmark

The impetus for the Retro Contest was a paper published by researchers at OpenAI [3]. The researchers proposed that the SEGA Genesis *Sonic the Hedgehog* series was an appropriate domain for measuring *cross-task generalization* properties of RL algorithms (essentially, how well said algorithms could use knowledge acquired from solving previous tasks to solve new ones). A select few of several characteristics that render the series suitable for these tests include:

- Agents may train on designated "training" levels and be evaluated on "test" levels
- Levels encourage exploration with multiple paths and unorthodox obstacles
- *Meta-learning* can be implemented by running multiple models simultaneously

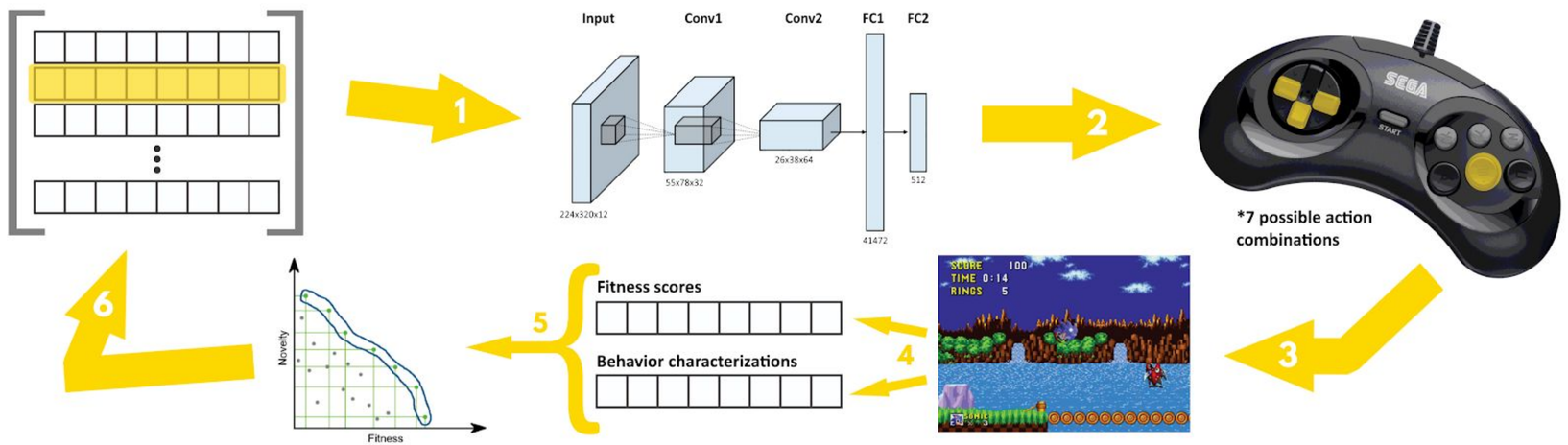
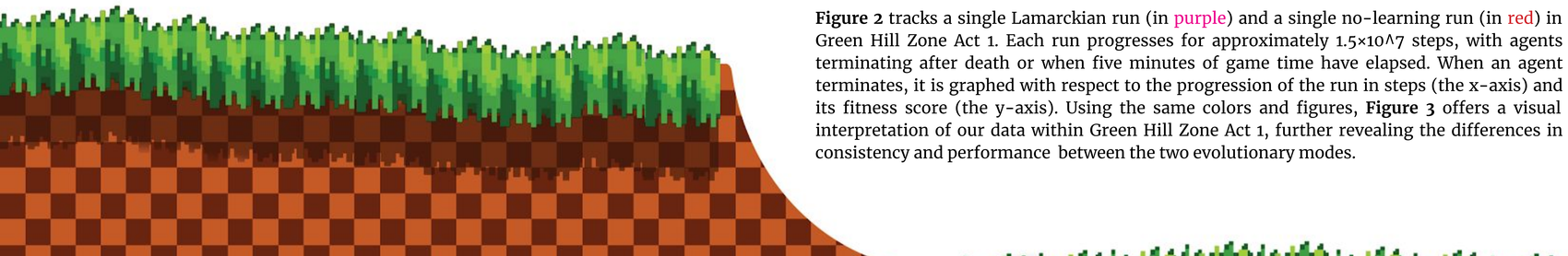


Figure 1. *Structure of our solution, which utilizes NSGA-II and a second objective on top of a PPO-based policy.*

Our solution

We combine OpenAI's Proximal Policy Optimization (PPO) algorithm [3], the Non-dominated Sorting Genetic Algorithm II (NSGA-II [2]), and a behavioral diversity objective [4]. We start with a set number of solution genomes, all of which contain randomly generated solution vectors. **Figure 1** displays the evolutionary process:

1. Solution at index is copied into the weights and biases of a Convolutional Neural Net.
2. CNN takes game frames as input and suggests one of seven possible actions to take.
3. Suggested action becomes the one Sonic takes in-game.
4. After learning to play for a fixed number of steps, a final evaluation generates a fitness score and a behavior characterization for the genome.
5. The behavior characterization defines a novelty score that is used with fitness to define a Pareto front of fitness vs. novelty.
6. NSGA-II determines the most "fit" solutions (i.e., those on the Pareto front of both objectives) which are used to produce the next generation.



Preliminary results

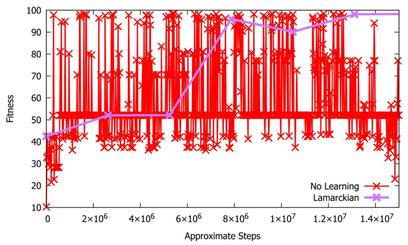


Figure 2. *Comparison of performance between no learning and Lamarckian.*

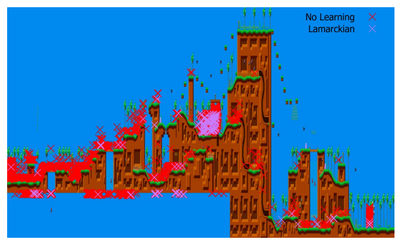


Figure 3. *No learning and Lamarckian modes' GHZ Act 1 end states graphed.*

Figure 2 tracks a single Lamarckian run (in purple) and a single no-learning run (in red) in Green Hill Zone Act 1. Each run progresses for approximately 1.5×10^7 steps, with agents terminating after death or when five minutes of game time have elapsed. When an agent terminates, it is graphed with respect to the progression of the run in steps (the x-axis) and its fitness score (the y-axis). Using the same colors and figures, **Figure 3** offers a visual interpretation of our data within Green Hill Zone Act 1, further revealing the differences in consistency and performance between the two evolutionary modes.

Conclusion

As displayed in the figures, no learning reaches the end of the level quickly, but it suffers from a grave lack of consistency. On the other hand, Lamarckian evolution slowly converges towards near-optimal performance, rendering it the more reliable of the two modes. The reinforcement learning aspect of Lamarckian evolution pushes it towards local optima, as shown in **Figure 3**. Lamarckian seems to converge around the loop in the middle of the level but is able to beat the level itself.

In conclusion, we show that despite no learning with behavioral diversity being able to discover an array of good solutions rapidly, learning is necessary for consistently honing in on the best solutions.

References

[1] Castillo, P.A.; Arenas, M.G.; Castellano, J.G.; Merelo, J.J.; Prieto, A.; Rivas, V.; Romero, G. Lamarckian Evolution and the Baldwin Effect in Evolutionary Neural Networks. Paper presented at: Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB); 2006; Tenerife, Spain.

[2] Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **2002,** *6*, 182-197.

[3] Nichol, A.; Pfau, V.; Hesse, C.; Klimov, O.; Schulman, J. et al. Gotta Learn Fast: A New Benchmark for Generalization in RL. Self-published by OpenAI; 2018.

[4] Mouret, J.-B.; Doncieux, S; Encouraging Behavioral Diversity in Evolutionary Robotics: An Empirical Study. *Evolutionary Computation* **2012,** *20*, 91-133.