Simulating Iterative Classifications/Clusters Through NeuroEvolution of Augmenting Topologies

Kenan Millet and Marcus Williams

New York, United States

Abstract

This paper describes the Grouping of Ordered Dendrites Evolving Networks using Fitness Functions (GoODENuFF) alogirthm, a neuroevoultionary algorithm designed to cluster objects through the generation of evolving neural networks. The algorithm is based off the exsiting NeuroEvoultion of Augmented Topolopgies (NEAT) algorithm and Clustering. We present an agent, GoODENuFF, which classifies objects based on neurual networking which identifies and reacts to them in the Infinite Mario Bros. benchmark. We claim that our Mario controller will have dynamically imporving preformance by (1) employing the concept of cell divison on varying genes, (2) promoting optimal actions or input events forward, and (3) iteratively dividing in game ojects into specified clusters. The paper describes the benefits of classifiying algorithms for game controllers.

Keywords: Dendrite, Cluster, Classification, Neural Network, Neuroevolution, Topologies

1. Introduction

Neuroevolution (NE), a branch of machine learning that uses evolutionary algorithms to train neural networks. Commonly used in computer games, NE has been adapted to tackle a variety of real-world challenges. In the case of video games evolutionary algorithms have been used to create controllers that play in the style of, human players (Karavoskiy and Togelius 2012). These controllers can be used both to guide players when they get stuck and to automatically test new game levels and features (Karavoskiy and Togelius 2012).

2. Background

In using NE algorithms such as NeuroEvolution of Augmented Topologies (NEAT), preforming benchmark problems like Infinite Mario Bros. have become increasingly more competitive in designing smarter AI. The NEAT algorithm is designed to take advantage of structure as a way of minimizing the dimensionality of the search space of connection weights in order to improve efficiency results from topologies (Stanley and Miikkulainen 2002). In using the Infinite Mario Bros. benchmark, we propose that we can make a 'good enough' Mario agent that uses NEAT as a means of clustering. By using NE, we plan to generate neural networks to group a set of objects, or entities into unique clusters. Our idea is to create a "smart" Mario AI that classifies entities it encounters in the game and make decisions on how to react to said entity. When humans play games they naturally create associations and grouping of the in game entities. In the case of Mario when a person sees a "Goomba" or a "Koopa" they classify them as enemies, something to avoid. Whereas when they see a mushroom or a fire flower they classify them as power-ups, something to seek out. The Grouping of Ordered Dendrites Evolving using Fitness Functions (GoODENuFF) is designed to do just that.

3. Good Enough

Genes

In the GoODENuFF algorithm, genes are represented as pairs of dendrites, which contain screen coordinates and a classification label, and axons, which contain a button press. When an object in the game (besides Mario) passes over a dendrite (i.e. the location of the object in screen coordinates is equal to the dendrite's screen coordinates), if the classification label of the object matches the classification label of the dendrite, then the axon that is paired with that dendrite is activated, resulting in the execution of the action (button press) that the axon is bound to.

Population Generation and Mutation

Each gene is represented as a dendrite/axon pair, a group of genes make up a network, and a group of different networks makes up a population. Each network is tested in a level, after which, the bottom 20% of the population are replaced with clones of the top 20%. Additionally, a random number of the top 20% are cloned a second time, and a random number of the bottom 20% (previously the bottom 20-40%) are removed, in order to introduce some variance of population size. After the cloning, every network in the population is mutated. This process involves modifying the screen coordinates, labels, and actions of the dendrite/axon pairs. Screen coordinates are mutated based on a Gaussian curve, resulting in having a high chance of moving little if at all. Labels and actions are more likely to be modified the more their corresponding screen coordinates changed.

Clustering and Classification Labels

Dendrites contain a classification label that represents the classification of object which will activate them when the location of the dendrite and the object are the same. These classifications refer to clusters of objects. Initially, all of the objects in the game belong to the same cluster. Every time Mario takes damage from an object, a counter for that specific object increases. Objects with higher counter values have a greater chance of being given their own separate classification with every new generation/mutation of the population of networks. This makes it so that, in the beginning, when the agent still has a lot to learn, there will be very few calculations, but as the agent becomes more robust, it gains room to refine its logic through further classification of objects via clustering.

4. Conclusion

This paper has described the GoODENuFF algorithm and its applications in the Infinite Mario Bros. benchmark. Using the implementations as described the algorithm was able to successfully beat four levels out of 20 trials. The GoODENuFF algorithm can potentially be applied to any platformer as it is designed to make no assumption at the start of the game and generate classifications overtime. Unfortunately, due to the framework that was used after each generation Mario was unable to properly pass on its knowledge to the next generation. In order for the algorithm the classification to effectively cluster the data an inexpensive and fully observable emulator that tracks all entities, player status and collisions is needed. The algorithm has shown that the GoODENuFF algorithm can in fact recognize its environment and generate networks. We hope to continue testing the algorithm on future frameworks. We are also considering possibly writing our own framework or potentially testing it on multiple emulators.

5. References

[1] Karakovskiy, S., and Togelius, J.(2012).The Mario AI Benchmark and Competitions.IEEE Transactions on Computational Intelligence and AI in Games,4(1), 55-67.

[2] Stanley, K. O., and Miikkulainen, R. (2002). Evolving Neural Networks through Augmenting Topologies.Evolutionary Computation,10(2), 99-127.

[3] Voorhees, E. M. (1986). Implementing agglomerative hierarchic clustering algorithms for use in document retrieval.Information Processing and Management,22(6), 465-476.