Artificial Life Project Proposal

Ho Ko(20891459)

12, October 2020

Introduction

An artificial product with life-like behaviors or properties can be recognized as an artificial life. There are two major methods in achieving the life-like results: learning as a group such as genetic algorithm(GA), and individual learning such as Deep Q-Network(DQN). Either one of the methods are more than enough to solve complicated problems, but in this project I will use both of them to mimic the reproduction and learning process of a typical human.

Motivation

Evolutionary algorithm mainly depends on the random initialized different individuals, the genetic and mutation mechanism. Which by inheriting the best performing individuals from last generation, the new generation have a decent probability of improving the result. However this is not how genetic work in real life, where offspring will not inherit what stores in their parents. Every individual have to learn the ability to solve complicated task from scratch.

In this project, I will combine genetic algorithm, partial supervised learning and other methods in reinforcement learning to simulate reproduction, learning from parent, and self learning that most animals do.

Background

Evolutionary Algorithm

Evolutionary algorithm basically mimics the process of evolution to find the best adapted individual as the solution. By natural selection from a pool of different individuals, the gene mechanism can inherit the properties from best performers in last iteration, and the newly created individual will have a higher chance in performance improvement. If the algorithm converges to an acceptable result, the problem will be solved by picking out the best individual as the solution.

Reinforcement Learning

Reinforcement learning optimizes the reward estimated or obtained by the agent. By interacting with the environment the agent updates the reward function, obtained new information and improve the overall stored knowledge. If the algorithm converges to an acceptable result, the problem will be solved by making decision base on the knowledge collected by the agent.

Methodology

There are two projects highly related and can support my points in this section that I did in the past.

1. Economy Simulation(Python3 + Processing3):
   By letting 100 individuals do tradings in an open market, the agents have to decide an appropriate price to sell their redundant resources and a reasonable price to buy the supplies that they need to consume in order to keep alive. The result of the economy simulation have the same phenomenon of inflation, deflation, closed-market and normal state just like the real world economy.

2. Creature Simulation(Java, footage: `https://rb.gy/yops7k`):
   By letting 100 creatures with three sensors compete in an open area, the agents have to survive by finding grass to eat or by eating other creatures. There are creatures that knows how to maintain food supply by rotating between different grass field to consume, and creatures that chase and eat others when detected nearby creatures.

I will use Javascript or Python3 in this project, where I am going to let around 10 to 20 individuals compete in a shared mine. The one that knows how to collect most resources from mining and stealing from others will be the best performer. The reason I reduced the number to about 15 individuals competing at the same time is because I am planning to simulate the whole scenario in a 2D-grid world about a size of 40x30, and individual are not allowed to stack at the same block.

The mine will consist of air block, dirt block, rock block and gold block. If the result is too satisfying maybe more levels of rock blocks will be included. There will be a rest time needed for the agent to take after collecting a dirt block, a rock block or a gold block. Dirt blocks or rock blocks that are collected can be used to replace an air block.

Each individual will have an independent brain, with a single layer perceptron(SLP) to responses to the neighborhood(3x3 or 5x5). The agent should be smart enough to do the task as that is what the brain structure of the agents in my Economy Simulation were designed. Feature extraction might be needed to include if SLP is not enough to handle the task.

New agents will be created by random initialization or by breeding of the top performers from last iteration.(GA) The gene will only decide the structure of the brain.(the number of neurons and the connectivity between them) No weight will be inherited from the parents, and every individual have to learn from 0.

The newly borned agents will have to pass a learning phase in a smaller environment, obtaining experience from their parents and their own interest.(partial supervised learning, partial reinforcement learning) The parents will always tell their kid what will they do instead after the kid made a decision during the learning phase, and the kid will have their own opinion on how well they are doing by predefined reward function. After every individual were trained, all the newly graduated kid will be start the real competition in the 40x30 shared mine environment, and keep learning by its own experience afterward.

There are a few factors not decided yet:

- Is the agent being killed after being stolen.
  - If this is true, the result might be every individual killing each other before start digging for gold.
  - If this is false, then the result might be every one waiting to steal the resources that other collected, so the stealing amount should only be a small fraction of the owner.
- Is the small environment(for learning phase) separated for each individual, or do they learn in a shared common area.
  - If it is shared, the result might be some individuals will have no gold left to mine and learn, as others are simply better and mined it before they can find it.
  - If it is not shared, competitor will not exists, and it will be impossible to learn how to deal with them. So fake competitors are needed, which usually will be biased.
- Should the number of iterations in learning phase for each individual be the same?
  - If it is the same, what will be that number? The reason to fix the number of learning will be separating the good genes from the bad one. As a good brain with correct number of dropout should have a more robust performance in learning. Although, the number will be hard to adjust as the result of the learning phase is hard to define.
  - If it is not the same, what should be the indicator that one is fully trained?

There will be more problems appearing after I start working on the project, as maybe some decided plans will be changed in the final result.

Evaluation

There are multiple results that can be evaluated. The increment of the best agent's resource score is the critical indicator of whether the agents are being more intelligent or not. By analysing the increment of resources obtained by the same bloodline, we can conclude if the parenting system works.(If knowledge can be transferred though teaching the next iteration) More experiment might have to be done in order to compare with no-parenting system scenario. Other results might exists, for example the strategy that most agents used, but they might be so unpredictable that I can not make any assumption yet.