Comparing Genetic Algorithms with Reinforcement Learning through OpenAI's RetroGym Suite

Abstract & Introduction

In this paper, we will compare the benefits and drawbacks to using Genetic Algos in video games. This will be the second option:

- Apply / extend existing algorithms to a new application / task / dataset.

If you choose to work on this of this category, you will need to implement and analyze the performance of **at least two** different deep learning algorithms / methods in a task domain, e.g., image recognition or natural language processing. You are asked to discuss the strength and weakness of each of the approaches backed by your experimental findings.

We plan to compare genetic learning algorithms which work well on static environments (in our case games such as Mario, F1Zero etc.) to reinforcement learning of the same games. Our main goal is to compare how well these algorithms generalize to other non-static games. Since OpenAI provides us the same interface to many games, we can apply the same framework without major changes in the base code. See the example below for how OpenAI interfaces an input to ANY game in its suite:

```
env = retro.make(game='F1-Genesis')
obs = env.reset()

once = True

actions = np.zeros(len(env.action_space.sample()))
while True:
    obs, rew, done, info = env.step(actions)
    print(obs)
```

You can see we can collect observations (game pixel data), assess rewards of any actions and tune them, as well as game metadata under "info" (for example number of cars on screen).

Related Works

1) https://flyyufelix.github.io/2017/10/12/dqn-vs-pg.html
    a) In this paper, they compare different reinforcement learning algorithms.
    b) "At present, the two most popular classes of reinforcement learning algorithms are Q Learning and Policy Gradients. Q learning is a type of value iteration method aims at approximating the Q function, while Policy Gradients is a method to directly optimize in the action space."[1]
2) https://medium.datadriveninvestor.com/which-reinforcement-learning-rl-algorithm-to-use-where-when-and-in-what-scenario-e3e7617fb0b1
    a) The following are the concluding points directly from the article above.[2]
        i) It can be observed that PPO provides a better convergence and performance rate than other techniques but is sensitive to changes.
        ii) DQN alone is unstable and gives poor convergence, hence requires several add-ons.
        iii) SAC is very efficient for energy based optimization techniques due to entropy factor regularization
        iv) TD3 and TRPO work well with continuous action spaces but lack the faster convergence rate

v) A3C is very useful when large computation power is available and concept of transfer learning on similar environments needs to be introduced.

3) https://openai.com/blog/openai-baselines-ppo/
    a) OpenAI's overview of PPO used on their platform and the improvements they make over other reinforcement learning algorithms

Method & Algorithm

Two approaches will be investigated:

1. Reinforcement learning: Q-learning
2. Reinforcement learning: Proximity Policy Optimization (OpenAI's default learning algorithm)

Additionally in the case where either of the above algorithms become too difficult to implement or study, we may remove one of them and in place use Genetic Learning Algorithm.

Summary

We plan on comparing two popular state of the art brand new very used in industry cutting edge top of the line algorithms: Proximity Policy Optimization and Q-Learning.

Q-Learning is one of the most common reinforcement algorithms which aims to learn the best rewards to give to specific actions based on the state of the environment. For this reason we see it to be perfectly adaptable to learn games such as Super Mario World.

On the other hand, OpenAI, which is the company that provides us with the platform to learn SNES games (GYM Retro), promotes Proximity Policy Optimization which they claim to be the most adaptable to many different games and solve many issues related to gradient descent such as learning rate.

In our paper we will closely analyze:

1) The focuses of each algorithm and how it extends to learning our games
2) The differences to how each algorithm tackles a portion of learning (such as optimizing learning rate in gradient descent)
3) Lastly how each algorithm performs on different types of games which may present different problems (platformers will learn very different features than something like a fighting game where our algorithms must compete against a primitive non-static AI)
4) Overall complexity analysis will be done on each algorithm as well

References

Yu, Felix. *Deep Q Network vs Policy Gradients - An Experiment on VizDoom with Keras*, Github.io, 12 Oct. 2017, flyyufelix.github.io/2017/10/12/dqn-vs-pg.html.

Tewari, Ujwal. *Which Reinforcement Learning-RI Algorithm to Use Where, When and in What Scenario?* 25 Apr. 2020, medium.datadriveninvestor.com/which-reinforcement-learning-rl-algorithm-to-use-where-when-and-in-what-scenario-e3e7617fb0b1.