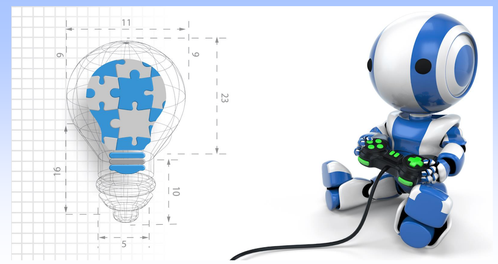
LEARN2PLAY

Anirudh Chaudhary, Quan Le, Richard More, Hao Nguyen

J. Mack Robinson College of Business - Georgia State University

CIS 8395 Big Data Experience – Spring 2020 - Professor Gandapoli



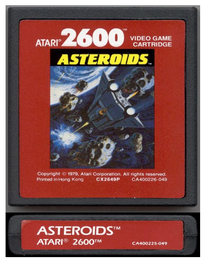
Introduction

OVERVIEW:

In this project, we are dealing with a paradigm of machine learning i.e. Reinforcement learning (RL) held the implementation potential in multiple disciplines, not limiting to, game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. It is useful to consider Reinforcement Learning as a possible solution when the problem statement deals with large state spaces or is looking to augment human behavior by providing decision support. Also, in areas where simulations are used to teach machines through trial and error while dealing with highly complex systems.

Our aim is to develop a system powered by Reinforcement Learning algorithm to achieve superhuman performance in a game and implement the learnings gained in fields like Robotics, Finance, Health & Medicine and Media & Advertising. We have decided to use the Open AI gaming platform and Atari 2600 console as the means to achieve the objective of the project. In the subsequent sections, we will be discussing about Reinforcement learning in detail, along with its types. After which, we will cover the architectural overview of the project form Reinforcement learning perspective as well as implementation perspective using AWS. Finally, we will discuss about the Reinforcement learning elements pertaining our project.

The Game: *Asteroids By Atari2600*



Actions

- Up
- Left
- Right
- Shoot
- Left + Up
- Right + Up
- Shoot + Up
- Left + Shoot
- Right + Shoot

Properties

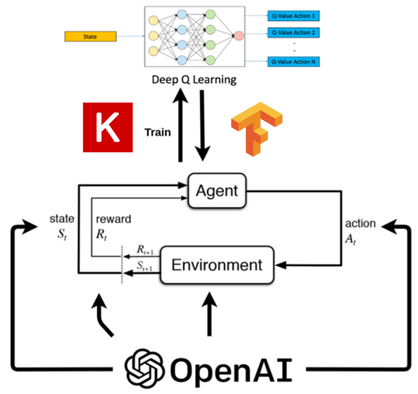
- Lives
- Scores
- Spaceship
- Asteroids

How to play:

- The player controls a single spaceship in an asteroid field which is periodically traversed by flying saucers.
- The object of the game is to shoot and destroy the asteroids and saucers, while not colliding with either, or being hit by the saucers' counter-fire.
- The game becomes harder as the number of asteroids increases.

Architecture

1. OpenAI provides Environment, Actions, States and Reward set up
2. Environment provide states and reward based on actions of Agent
3. Agent takes state as input to generate Q value for each possible action
4. Agent build DQN architecture by either TensorFlow or Keras library
5. Final action will be provided to environment based on highest Q-value action



Agent Setup

Keras RL

- Policy: LinearAnnealedPolicy
- Steps: 50,000 (~ 20 episodes)
- Neural Net Layers:
  - 1 Input, 1 hidden layer (Dense), 1 output
  - 1 Input, 5 hidden layers (1 permute, 2 Conv2D, and 2 Dense),1 Output
- Metric: Mean Absolute Error
- Optimizer: Adam
- Gamma: 0.99
- Warmup steps: 1000

TensorFlow Agents

- Agents
  - Reinforce
  - DQN
  - C51
- Optimizer: Adam
- Hidden layer structure per experiment
  - 100 nodes
  - 200, 64 nodes
  - 1000, 500, 50 nodes
- Longest training
  - 2,000 episodes
  - 4,000,000 steps

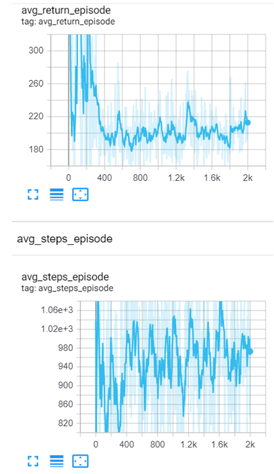
Results

Keras RL

At episode 17, the agent is can complete episode with 1610 point and half of total step. In this episode, agent can chase asteroid and identify the location asteroids. The only problem with agent now is that it can't dodge asteroids if it flies in discretionary distribution. Overall, Keras RL model can perform the best in term of score and number of training steps.

TensorFlow Agents

After more than 2000 episodes of training the agents have stabilized themselves too with the lowest scores they could reach. After generating the footage using the policies, we could verify the hypothesis that including a "stay" action resulted in less useful actions to be performed and most of the times the agent chooses to do nothing.



Conclusions

- Reinforcement Learning (RL) is data hungry
- RL project needs hardware with high configurations
- Using cloud for RL is not cheap
- RL is hard to predict model training time
- RL is not suitable for mission critical applications

Challenges

- AWS has some limitation doing RL
- Cost of training RL is very high
- RL require a lot of time to train
- Hardware utilization may or may not supportive to some technologies
- Hyperparameter combinations required many trial to find out the best
- Understanding Reinforcement Learning in a short period of time is difficult

References

1. https://medium.com/analytics-vidhya/reinforcement-learning-and-game-theory-28208ee795f8
2. https://upload.wikimedia.org/wikipedia/commons/thumb/9/93/Amazon_Web_Services_Logo.svg/1200px-Amazon_Web_Services_Logo.svg.png
3. https://hyperspin-fe.com/siteuploads/downloads/screenshots/monthly_10_2016/d802aff83bfcd5a62645c30aba35388a-atari-2600-alt.png
4. https://www.mobygames.com/images/covers/l/34133-asteroids-atari-2600-media.jpg
5. https://openai.com/content/images/2019/05/openai-cover.png
6. https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/
7. https://paperswithcode.com/sota/atari-games-on-atari-2600-asteroids