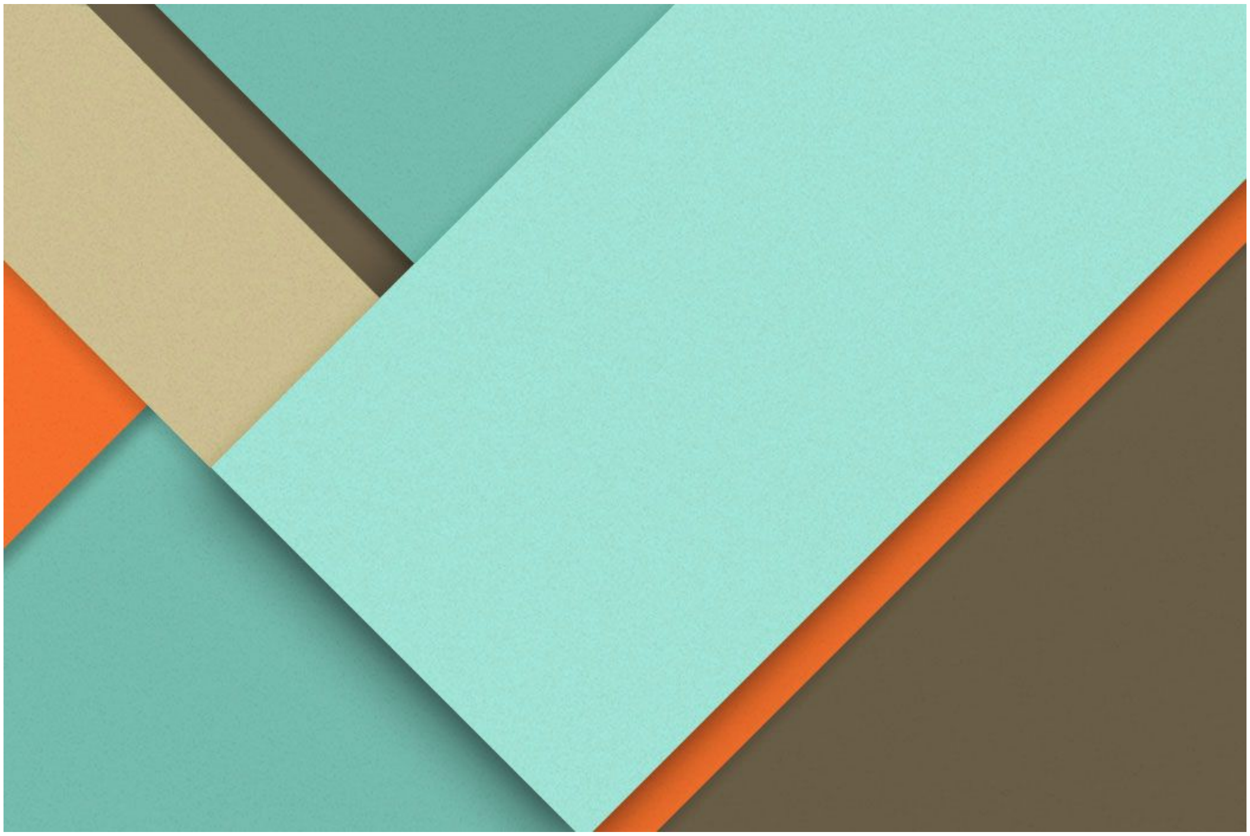

Berzerk playing Intelligent Agents

Devavrat Kalam
Rochester Institute of Technology
611 Park Point Drive
Rochester, NY 14623

Introduction

Artificial Intelligence is a vast field with all sorts of applications ranging from simple prediction systems to deep neural networks inside brain for robots. It can also be used for playing video games. We might be familiar with some of the most famous AI agents ever created like IBM's Deep Blue who beaten world champion Garry Kasparov in game of Chess and DeepMind's AlphaGo who beaten world champion Lee Sedol in game of Go.

In this project, we will be implementing simple reflex agents who plays a classic Atari game named "Berzerk". The game rules are pretty simple, man travels from one room to another shooting robots while dodging their attacks. The game environment is taken from OpenAI's Gym library. The library contains 100's of game environments ranging from old classic Pac-Man to advance games like GTA.

The objective of the Berzerk playing agents -

- Score highest as possible,
- Complete most of the levels,
- Shoot many robots as possible,
- Finish stages in lowest time as possible

I have implemented total three agents, one random action agent and two reflex agents. The performance measures are based on the objectives stated above. Analysis is done by represented histograms, box & whisker plots and performed t-tests to differentiate the data sets.

Important Definitions:

- Episode: Duration for man to finish the game or loss all the lives (typically 3).
- Level: Environment count. When man go from one room to another or dies, background changes. That is counted as level increase.

Methods

Different Agents and their method of implementation -

1. Random agent -
    - Random agent is the simplest agent who selects actions in random manner among all possible action set in any condition.
2. Agent1 -
    - Agent1 is a reflex agent.
    - It first detects man, wall and robot pixel coordinates as (x,y) tuples.

- If there is a robot in any four directions (up,down,left,right). If present, immediately shoot.
- If any wall or bullet from enemy is approaching in close proximity, It will move away from that position to safeguard itself.
- Finally, if none of above applies, it will try to move to another room from top exit.

3. Agent2 -
    - Agent2 is also a simple reflex agent.
    - It first detects man, wall and robot pixel coordinates as (x,y) tuples.
    - If there is a robot in any four directions (up,down,left,right). If present, immediately shoot.
    - If all enemies are dead, it will try to move to another room from top exit. If not, it will select random direction from possible directions at that instance (avoiding walls) for movement.

Results

Results are calculated for every agent over 1000 episodes each.

Random agent:

- Maximum score obtained = 650
- Total time for 1000 episodes in seconds = 409.42
- Total time for 1000 episodes in minutes = 6.82
- Average time per episode in seconds = 0.41
- Average number of enemies killed = 3.3
- Average score per episode = 167.46

Agent1:

- Maximum score obtained = 2430
- Total time for 1000 episodes in seconds = 5381.58
- Total time for 1000 episodes in minutes = 89.69
- Average time per episode in seconds = 5.38
- Average number of enemies killed = 16.69
- Average score per episode = 948

Agent2:

- Maximum score obtained = 2040
- Total time for 1000 episodes in seconds = 3929
- Total time for 1000 episodes in minutes = 65.48
- Average time per episode in seconds = 3.93

- Average number of enemies killed = 12.4
- Average score per episode = 697.15

Other detailed analysis in Histograms, Box & Whisker plots and t-test is performed in the provided supporting Jupyter file named “analysis.ipynb”.

Conclusion

A good intelligent agent should be programmed based on the objectives it has. When it is built, it should be tested several times to check what are the anomalies and should be modified accordingly. Finally, the code should be optimized to increase speed and efficiency which is very crucial and yet most underestimated part of any program.

Random agent is like a monkey with knife. Nothing good can come out of the monkey-knife pair. In a nutshell, creating reflexive agent is easy and great learning experience for anyone who is a beginner or a professional in this field.