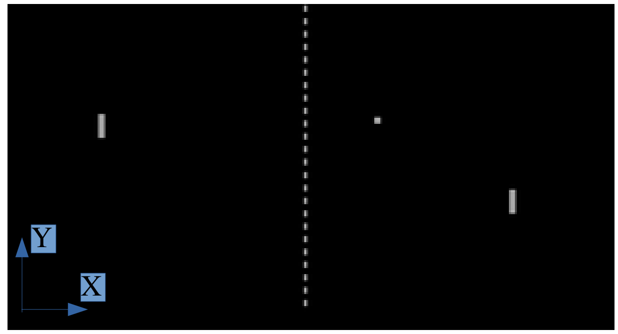
Autonomous '*pong*' player-agents

Authors: Rodrigo Arias <rodarima@gmail.com>
Cedric Bhihe <cedric.bhihe@gmail.com>

MIRI / ML Project Proposal
Delivery: before 2018.04.09 – 23:55

This proposal describes the study of a closed system consisting of two independent and autonomous ML agents, playing a game of *pong*[i] where each player-agent must overcome its opponent by learning better and faster in order to score points.



The game

The game consists of a rectangular arena (in the XY plane), a ball, and two paddles to hit the ball back and forth across the arena. A player-agent represented by a paddle scores when the ball flies past the opposite player's paddle and hits the back-wall opposite the scoring player's side. When this occurs a new exchange starts. Each player can only move vertically (i.e. along direction Y). The ball can bounce off the paddles as well as the side walls running parallel to axis X.

Proposal

We choose to focus not on the implementation of the game[ii], but rather on that of the ML methods we propose to study. That includes implementing two autonomous and independent ML agents to control each player's moves. By endowing each player-agent with different characteristics and parameters, our aim is to compare the relative performance of different learning methods along with their parameter settings. Given the nature of the problem, we choose to focus on reinforced methods[iii] such as off policy, model-free Q-Learning (QL), on-policy State-Action-Reward-State-Action (SARSA) and Deep Q Neural-Networks (DQN), without any guarantee at this early stage that we include every nethod in our final results.

We may introduce incremental complexity in the decision-making conditions of successive games so as to make learning by the player-agents gradually more difficult. To illustrate this, we only cite:

- paddles with mass to constrain its dynamics.
- paddles' exclusion zones in the areas defended by the players. This constitutes a deliberate vulnerability in the line of defense of each player, which a trained opponent must learn to recognize before taking advantage of it.
- non uniform ball bouncing off the surface of paddles to introduce variety in the trajectories of the ball.
- normally distributed noise in the positioning of the paddle

Performance comparison will take the form of tournaments between players, where the winner will proceed to the next level to face a newly configured player-agent.

Motivation

Q-Learning, SARSA and DQN adapts particularly well and in different ways to incomplete knowledge situations and never-seen-before states, where - as is our case - player-agents are memoryless and rely on a limited set of actions to maximize their utility. Each player's action is rewarded by a numerical score, whereby a player-agent learns what the optimal action is in any given state. Q-Learning is about how to perform best in an _a-priori_ largely unknown environment.

'Pong' has two players, and affords us the possibility to either pit one learning agent against another or to appraise its learning curve when opposed to a human player. This permits the design of a parametric study of learning performance, as a function of reward, discount factor and learning rate or "step size", where two (possibly unequally parametrized) learning agents compete in gradually more complex learning environments.

Conclusion

Apart from the simplicity of the game's rules and the fact that the game has already been implemented many times in a variety of languages, among them Python, Java, C y C++ (to the best of the present authors' modest knowledge) , there are several ML related reasons to choose a game such as `pong` to study the performance of different reinforced learning method.

---

i The game of 'pong' is one of the earliest video games, released in 1972 by Atari. It is built with simple 2D graphics.

ii See https://trevorappleton.blogspot.com/2014/04/writing-pong-using-python-and-pygame.html for an example of implementation using Python.

iii R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction., MIT Press, Cambridge, MA, 1998.