Fun with Pong

Cédric Bhihe and Rodrigo Arias

April 9, 2018

Introduction

Pong is one of the earliest video games, released in 1972 by Atari. Is a 2-players table tenis sport game built with very simple 2D graphics.

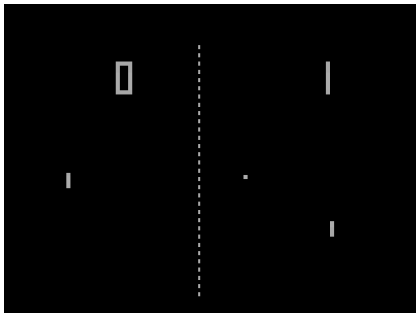

Figure 1: Original Pong game

Each player controls a paddle by moving it vertically across the table, in order to hit the ball. The objective of the game is to make the opponent miss the ball. Each ball missed scores a point to the opponent, and a new exchange starts. The ball can bounce off the paddles as well as the side walls.

Proposal

We propose the design and implementation of player controller agents by using different machine learning methods. The agent should be able to move the paddle as a human would do. The objective of the project is to compare the performance of various ML agents against one another. In order to test them a tournament is played, such that the winners will play against each other until there is only one winner.

Given the nature of the problem we choose to focus on reinforcement methods like Q-Learning [1]. Different agents can be either complete different RL methods or only with small differences in the parameters.

We may introduce incremental complexity in the successive games so as to make learning by the ML agents gradually more difficult. Some examples may include:

- Paddles with mass to constraint the moves.
- Holes in the space covered by the paddles, which a trained player could exploit.
- Non-uniform bounces in the paddles to modify the trajectory of the ball.

Motivation

One of the main reasons to choose a game like the Pong, is the simplicity. We can focus on different methods, so we can learn without spending a lot of effort in the implementation. The game is very common and likely to be already implemented, so we can reuse parts of it.

As we control the problem, we can adjust the complexity of the game as far as we want. So we are not fixed by a complex problem, but a simple one that we—as humans players—already know how to solve. Reinforcement methods can deal with incomplete knowledge problems, which are very common in the real life. Once we understand how to apply them to a simple problem, we can continue to more complex problems.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.*, MIT Press, Cambridge, MA, 1998.