

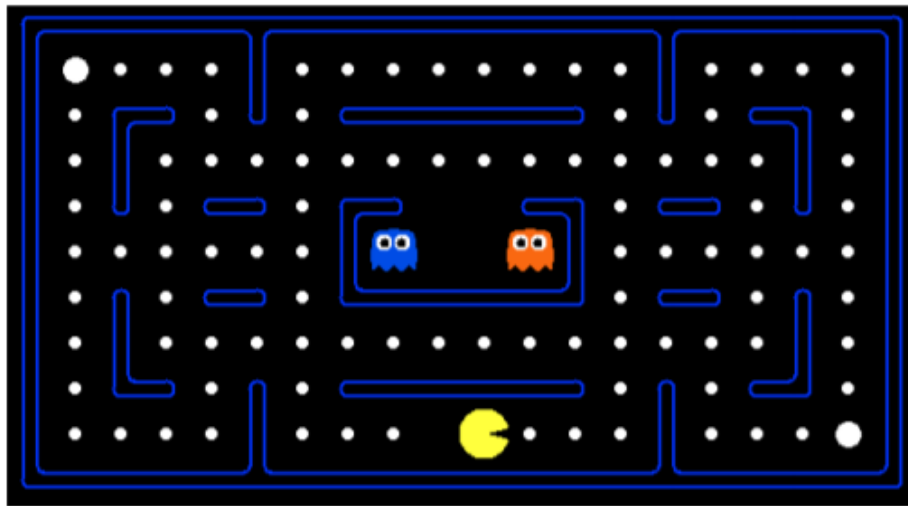
# Implementing Deep Q Learning and Double Deep Q learning in training Pac-Man

Final Project: Project Proposal

ECE 517 - Reinforcement Learning  
Prof. Dr. Amir Sadovnik

Jerry Duncan, Fabian Fallas, Tabitha Samuel

November 2020



# 1 Problem definition

Using reinforcement learning techniques to teach algorithms to ‘play’ Atari games has been a popular and constantly evolving research area in the past five years. For this final project, we propose to use Deep Q-Learning, and double deep Q-Learning reinforcement learning techniques to train a model to successfully play the Pac-Man game. We will follow Gnanasekaran et al. [1] in condensing the Pac-Man grid into a  $n \times m$  image where each pixel represents a possible state (e.g. one color will represent walls, another the ghosts, etc). We will initially train the models on the small grid provided by [2], and then compare change in complexity and run-time when they are run on the medium sized grid.

# 2 Problem motivation and background

Most research problems are usually supervised tasks that need the regular pipelines of training and testing. Also, traditionally these tasks are applied over environments that are “static”, that is, the data helps to tune the weights and the resulting model can be applied over the validation set. But with reinforcement learning we can apply algorithms to dynamic and evolving environments and, as in the case of video games, the artificial intelligence can be seen in action - in a way that is not just responding to a static environment. Consequently, we want to use video games in order to experience how an agent can play the game without human intervention. By doing this project we will face the following challenges:

- Program our first function approximation algorithms.
- Use an interface for video games.
- Use Deep Q-Learning which is a well-noted solution for complex tasks.
- Understand and program Double Deep Q-Learning.

We propose to build upon the work described in [1]. In this paper, the group investigates the effectiveness of Deep Q-learning based on the context of the Pac-Man game using Q-learning and Approximate Q-learning as baselines. We plan on implementing Deep Q learning, and adding an additional model with Double Deep Q learning, and comparing performance and accuracy trade offs between the two techniques. We also plan on adding prioritized experience replay to see if we can increase the speed in which our models learn, using both [3] and some of our own intuitions of which experience to replay.

# 3 Evaluation of results

We would like to evaluate results from three perspectives in this project:

- Evaluate increase in complexity and training time when we move from the small grid to the medium sized grid.
- Compare performance of Deep Q-learning versus Double Deep Q-learning. By using Double Q-Learning we will implement something different from [1].
- Attempt to add prioritized replay in the way described by [3] and separately using some of our own ideas.

If time permits we would like to create an implementation of regular Q-Learning. This can be interesting since by using this implementation uses a simpler function approximation.

## References

- [1] A. Gnanasekaran, J. F. Faba, and J. An, “Reinforcement learning in pacman.” [Online]. Available: <http://cs229.stanford.edu/proj2017/final-reports/5241109.pdf>
- [2] UC Berkeley CS188 Intro to AI. [Online]. Available: [http://ai.berkeley.edu/project\\_overview.html](http://ai.berkeley.edu/project_overview.html)
- [3] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, “Prioritized experience replay,” 2016.