Project Proposal: Deep-Learning implementation in Pac-Man

Introduction

Machine Learning, associated with Artificial Intelligence, is now vastly used in the industry, including applications in medical diagnosis, electronic trading, fraud detection, online search, and so on. With that observation, we can ask ourselves if Machine Learning and Artificial Intelligences could, one day, replace human-computer interaction completely in the most trivial applications like, for instance, playing a video game.

Statement of Problem

Creating an Artificial Intelligence (AI) itself doesn't make it "intelligent", a Machine Learning algorithm needs to be implemented to train our AI.

In the case of a video game, it's unlikely to have one solution or technique that fits all situations.

In the event of building a Supervised Learning-based Machine Learning algorithm, the amount of data required to build an efficient AI will be massive and more likely to omit some game mechanics.

Objectives

I propose to build an Artificial Intelligence, trained by a Machine Learning algorithm, that will meet the following criteria:

(1) Deep Reinforcement Learning (DRL) and Genetic Algorithm: The algorithm will iterate repeatedly inside the defined environment to train itself (doesn't use a pre-constructed training dataset) by using operators such as mutation, crossover, and selection on a population of genes, representing the nodes, strength values, and bias inside an Artificial Neural Network (ANN).

(2) Retrieve inputs for our Neural Network from an open-source Pac-Man that will follow the same rules (behavior of the entities) even if a new level is introduced.

Deliverables

At the end of the project, I expect to have a program that will implement the requirement above, as well as:

(1) Tools that shows the progress of the Genetic algorithm breed, such as the score of the fittest Neural Network and the average score of all of them during the training.

(2) Trained Neural Network for demonstration that can beat, at least, the level of Pac-Man used for the training iterations and a completely new level where it was NOT trained.

Resources

This program will be created using Python 3.6 (or newer) and the libraries TensorFlow, because of its huge community and faculties for Deep Learning, and Keras, used an add-on for fast-experimentation in Deep Neural Networks.

The Pac-Man game will be retrieved from an open-source repository on GitHub that will fit our project requirements.

Sources

https://en.wikipedia.org/wiki/Genetic_algorithm

https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_fundamentals.htm

http://codeinpython.com/tutorials/deep-learning-tensorflow-keras-pytorch/

https://deeplearning4j.org/compare-dl4j-tensorflow-pytorch

https://en.wikipedia.org/wiki/Neuroevolution_of_augmenting_topologies