# Machine Learning Engineer Nanodegree

## Capstone Project

André Tadeu de Carvalho October 8th, 2017

## Proposal - Learning to play video games with Machine Learning

### Domain Background

According to (https://www.mpaa.org/wp-content/uploads/2016/04/MPAA-Theatrical-Market-Statistics-2015_Final.pdf) and (https://businesstech.co.za/news/lifestyle/88472/the-biggest-entertainment-markets-in-the-world/), the movie industry reached $38.3 billion in revenues in 2015, whereas the video games industry reached $91.5 billion in revenues in the same period, which is more than the double of the theatrical market revenues in the same period, and it is projected to reach $107 billion this year. These facts suggests that there still some markets to explore or to improve in existings markets. For example, the advent of esports, in which participants compete in multi player games, such as Dota 2, Counter Strike, and League of Legends create an opportunity to develop agents that can learn from the environments, from humans, and even from Adversarial Networks, to create AIs for training player for esports, or even competing against them. One concrete effort in this direction is that DeepMind and Blizzard are working together to use Deep Reinforcement Learning to create an agent that plays Starcraft II, see more (https://deepmind.com/blog/deepmind-and-blizzard-open-starcraft-ii-ai-research-environment/) and (https://github.com/deepmind/pysc2).

I believe that Reinforcement Learning can be use to create more entertaining AIs for multi player games, which extends to single player games. As the CPUs gets faster and the GPUs can be used to train Neural Networks, it is feasible to train AIs to be smarter than the current ones that rely on scripts with simple hard-coded strategies.

### Problem Statement

In this Capstone project I will create an agent for the Atari game Breakout, since it is a project with educational purpose and with stricter set of resources to use. Breakout is a game which its objective is to destroy all of the bricks on the top of screen. To do so, the player controls a paddle and use it to bounce a ball, which will bounce back to the top of screen, hit the bricks, destroying them. The player wins whenever all the bricks are destroyed and it loses whenever all the balls that the player has available goes to the bottom of screen. I will use the Breakout-v0 environment from OpenAI Gym.

(Image from (http://blogtectoy.com.br/conheca-breakout-o-classico-jogo-desenvolvido-por-steve-jobs-para-atari/) )

## Datasets and Inputs

Reinforcement Learning problems generally does not employ external datasets and Breakout does not requires one. The input of the game, according to (https://gym.openai.com/envs/Breakout-v0/), is each frame that is drawn on the screen. Each image is an RGB image with 210 x 160 pixels, that can be mapped to an array of (210, 160, 3) dimensions. The actions is performed repeatedly for a duration of *k* frames.

All this data can be generated during the process of learning, so I need not play the game to generate data to the agent and perform a subsequent supervised learning from this data. The agent should be learning to play the game by playing for itself.

## Solution Statement

To solve this problem, I will employ Neural Networks to map *k* frames and all possible actions to rewards and then combine it with Q-Learning, Policy Gradient or other method in Reinforcement Learning literature to obtain the best performance between the models. The following book will be my guide in this task: (http://incompleteideas.net/sutton/book/bookdraft2017june19.pdf). Q-Learning was taught in the course and follows the formula Q(s_t, a_t) = Q(s_t, a_t) + \alpha * (r_t + \gamma * max_a Q(s_{t+1}, a) - (s_t, a_t)). Double Q-Learning might be chosen, as explained in this paper (https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/download/12389/11847).

## Benchmark Model

To compare how well my model worked on Breakout, the evaluations from OpenAI Gym comes to aid me (https://gym.openai.com/envs/Breakout-v0/). Even though I am not comparing to a human, I am comparing with the best models available online. Comparing with a human would require to compare with the scores of the best human player of Breakout, which I am not and I do not have access to.

## Evaluation Metrics

The evaluation metric is straightforward for this game: as far as I know it is just the scoreboard.

## Project Design

For this project, I intend to work with OpenAI Gym platform, whose source code is here, in which is possible to implement the models with either TensorFlow or Theano. OpenAI Gym already contains the environment for Breakout, called Breakout-v0 (as previously cited).

For the Neural Network part, I would have to experiment with several settings before deciding for the definitive architecture. Fully connected layers might be used, also Convolutional layers, and pooling layers. The gradient descent optimization algorithm used will be evaluated which one performs the best, from standard stochastic gradient descent to any optimizer that does not requires a batch of data to perform well (here are some of the optimizers (http://ruder.io/optimizing-gradient-descent/)).

For the Reinforcement Learning part, I am going choose one algorithm from either Policy Gradient, Deep Q-Learning, or Double Q-Learning. I intend to play a bit with the algorithms in order to examine how well they perform and choose the best performing one.