LearnyMcLearnface – Proposal (Draft)

* Albert Anguera Sempere (albert.anguerasempere@uzh.ch)
* Dominik Bucher (dominik.bucher@uzh.ch)
* Michael Ziörjen (michael.zioerjen@uzh.ch)

This is the proposal for the project of the group **LearnyMcLearnface** for the Reinforcement Learning course in the fall semester 2020.

1. Problem

We are going to tackle the 'CarRacing-v0'[[1]](#footnote-1) problem of OpenAI by Oleg Klimov. It is a simple continuous control problem. The environment is already provided[[2]](#footnote-2):

* States consisting of 96x96 pixels with RGB dimensions (discrete 0 to 255)
* Actions consisting of “steer”, “accelerate”, “brake” (continuous -1 to 1 respectively 0 to 1)
* Rewards -0.1 every frame and +1000/N for every tile visited with N being total number of tiles in track
* Game is solved when the agent gets 900+ points consistently
* Episode finishes when agent visited all tiles or leaves boundaries

The car racing problem can be discretized.

# Proposed Solutions

## (Deep Deterministic) Policy Gradient Methods

Policy gradient method might be a place to start. Make it grayscale using scikit-image[[3]](#footnote-3). Discretize the action space. Bigger chunks (ca. 100) at first rather than smaller chunks. Start with 10 steering, braking, accelerating choices.

-> Look at pong example in slides (5th), look at package provided for neural networks (4th lecture)

-> not do a convolutional neural-net, go for vanilla-multi-layer network and see how that works

## 2.1 Deep Q Learning

We are going to use a neural network to approximate the Q table. Could be simplified by outputting either steering or accelerating / braking (basically telling the neural net, not to break and accelerate at the same time).

# Question

* Is it possible to do model-based approaches? ->
* If we all have no prior experience with RL should we try to solve a simpler problem?

1. https://gym.openai.com/envs/CarRacing-v0/ [↑](#footnote-ref-1)
2. https://github.com/openai/gym/blob/master/gym/envs/box2d/car\_racing.py [↑](#footnote-ref-2)
3. https://scikit-image.org/docs/stable/auto\_examples/color\_exposure/plot\_adapt\_rgb.html [↑](#footnote-ref-3)