Model for Car Racing in OpenAI Gym using Reinforcement Learning

Harsh Agrawal
Department of Computer Science)
North Carolina State University
Raleigh, USA
hagrawa@ncsu.edu

Isha Pandya
Department of Computer Science)
North Carolina State University
Raleigh, USA
ipandya@ncsu.edu

Sanjana Kacholia
Department of Computer Science)
North Carolina State University
Raleigh, USA
skachol@ncsu.edu

I. Motivation

Automation is prevalent in all fields today and self-driving cars is one popular application. However it is not affordable to test such models in real time; since failure may have fatal consequences like crashes.Thus, a virtual environment is what this project provides. The goal is to achieve maximum accuracy for the car to trace a given path in the least possible time.

II. Challenges

1) Cannot train in real conditions as failure might have fatal consequences
2) The present algorithm does not take into consideration the presence of other agents(eg. traffic)
3) More data needs to be collected to reliably train the machine learning algorithms
4) In order to navigate down a road, the self-driving car requires very precise maps of the street and surroundings.Thus the present simulated version cannot be used to replace drivers as of yet.

III. Data

The data in this project is created using a simulator. We will be using CarRacing-v0 environment of OpenAI Gym to feed the data into out model.
It is a top down racing environment where the state consists of 96x96 pixels.

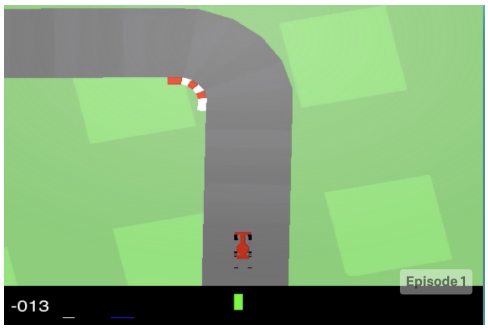

Fig. 1. Frame from CarRacing-v0

IV. Evaluation

The model describes actions(forward,right,left,back) and aims on training the car to predict the next move based on the environment variables. Desired outcome will be the car being able to trace the entire path and complete a lap. Next, we would like to improve the speed required for the same i.e. to minimize the number of frames required to complete a lap. The main focus here is to validate the actions predicted by the car to match with the actual path. Evaluation factors in focus are:
1) Portion of path traced correctly(maximize)
2) The speed of the car(maximize)
3) Number of wrong moves taken by the car(minimise)
4) Number of different paths vs accuracy improvement

References

[1] Ian Goodfellow, Yoshua Bengio, Aaron Courville, “Deep Learning“
[2] https://deepmind.com/research/publications/playing-atari-deep-reinforcement-learning/
[3] ttp://sitn.hms.harvard.edu/flash/2017/self-driving-cars-technology-risks-possibilities/