Lane-Keeping Using Reinforcement Learning

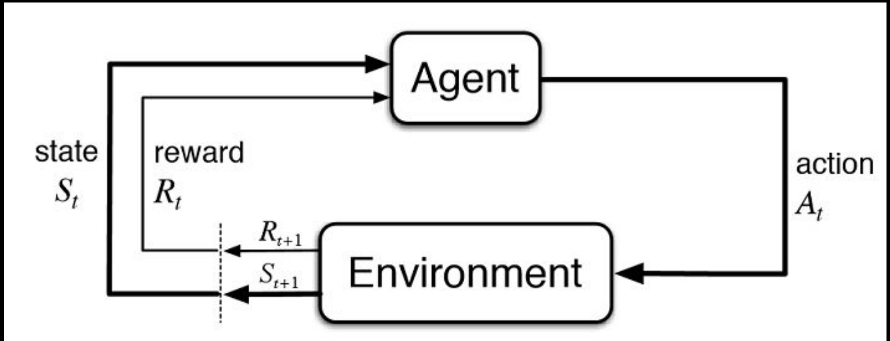
Ahmed Wael - Maha Ezzat - Nadine Amr

The Problem and Its Importance

- **Self Driving Car Task :**
    - Detecting lane.
    - Following the lane.
    - Maximizing the speed without collision.
- **Our Objective :**
    - Training a simulated autonomous vehicle using a reinforcement learning algorithm.

Learning Approach

Reinforcement Learning



Q-Learning

- model-free
- off-policy
    - epsilon-greedy policy

Q-Learning

$$Q^*(s,a) = R(s,a) + \gamma \max_{a'} Q(s',a')$$

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha Q^*(s,a)$$

Q-Learning

Hyper-Parameters to Tune:

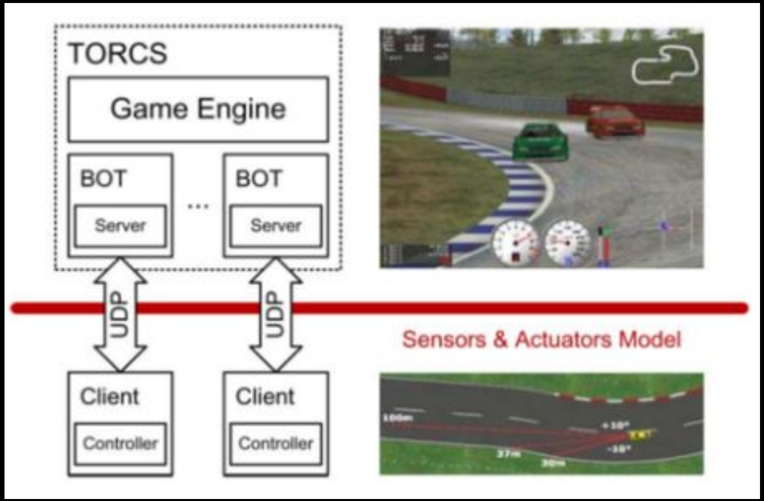
- ε: exploration rate
- γ: discount rate
- α: learning rate

The Open Racing Car Simulator - TORCS

Uses

- ordinary car racing game
- AI racing game
- research platform
    - human-assisted algorithmic generation of tracks
    - developing intelligent control systems for different car components
    - developing complex driving agents

SCR - Plugin



TORCS
Game Engine
BOT
Server
BOT
Server
...
UDP
UDP
Client
Controller
Client
Controller
Sensors & Actuators Model

TORCS Sensors

- Angle
- CurLapTime
- Damage
- distFromStartLine
- distRaced
- Fuel
- Gear
- lastLapTime
- racePos
- rpm
- speedX
- speedY
- track
- trackPos
- wheelSpinVel

TORCS Control Actions

- Accel
- Brake
- Gear
- Steering
- Meta

Adapted Methodology

States

- Sensors Used:
  - Speed
  - Angle
  - Track Position
  - Track

- Discretized to 11 Bits
  - 4 bits: speed sensor
  - 3 bits: number of sensor with maximum reading
  - 4 bits: average of 5 distance sensors

Action Selection

Steer	Acceleration(1)	Neutral(0)	Brake(-1)
0.5(Left)	0	1	2
0.1(Left)	3	4	5
0	6	7	8
-0.1(Right)	9	10	11
-0.5(Right)	12	13	14

Reward Function

- The car is out of track : It takes reward = -1.
- If the car is stuck : It takes reward = -2, and meta control action = 1 so the episode restart.
- If the car is neither stuck or out of track : It takes reward as a combination of the track position, the angle, and the distance travelled. These three factors add to 1.

Q-Table Update

- If the current state does not exist in the table, we create it in the Q-table with initial values = 0 for all actions.

- The Q-values of the previous state are updated using the current state, the previous state, the action, and the reward.

Progress So Far

- **Scenario 1**
  - ❑ Learning Rate = 0.01
  - ❑ Epsilon = 0.2
  - ❑ Discount Rate = 0.99
  - ❑ Reward

    - TrackPos<=0.75
      Reward = Rspeed+RtrackPos+Rangle
    - # 0.75<TrackPos<0.98
      Reward = 0.5*(Rspeed+RtrackPos+Rangle)
    - #TrackPos<=0.98
      Reward = -1.5
    - #Stuck
      Reward = -2

```
Rspeed=numpy.power((speed /float(160)),4)*0.05
Rtrackpos=numpy.power(1/(float(numpy.abs(trackpos)) +1),4)*0.7
Rangle=numpy.power((1/((float(numpy.abs(angle)) /40)+1)),4)*0.2
5
```

- **Scenario 2**
  - ❑ Learning Rate = 0.5
  - ❑ Epsilon = 0.2
  - ❑ Discount Rate = 0.9
  - ❑ Reward

    - TrackPos<=0.75
      Reward = Rspeed+RtrackPos+Rangle
    - # 0.75<TrackPos<0.98
      Reward = 0.5*(Rspeed+RtrackPos+Rangle)
    - #TrackPos<=0.98
      Reward = -1.5
    - #Stuck
      Reward = -2

```
Rspeed=numpy.power((speed/float(160)),4)*0.05
Rtrackpos=numpy.power(1/(float(numpy.abs(trackpos))+1),4)*0.8
Rangle=numpy.power((1/((float(numpy.abs(angle))/40)+1)),4)*0.1
5
```

Results

Measurement Results	Q-Learning Algorithm Scenario 1	Q-Learning Algorithm Scenario 2	Q-Learning Algorithm Reference
Training Episodes	1800	1800	1500
Number of Episodes to Complete one lap	------------------------	232	1200
Number of Episodes to Build the Model	------------------------	------------------------	1500
Maximum Speed of the Car	90 km/h	120 km/h	160 km/h
Time to Complete One Lap	------------------------	3:02:06	1:24:09

Future Work

Future Work

- More Models with different hyper-parameters.
- We suggest that a Continuous model will be build to compare the results and enhance the performance.
- Deep RL Methods as DDPG (Deep Deterministic Policy Gradients).

Questions

References

- K. Ibrahim, M. Ali, K. Mohamed, R. El Sayed and M. Farghaly, Lane Keeping Assistance of Autonomous Vehicles using Reinforcement Learning and Deep Learning on ROS. Cairo, 2018.
- A. Davies, P. Martineau, M. Molteni, A. Watercutter, M. Simon, E. Pao and L. Mallonee, "What Is a Self-Driving Car? The Complete WIRED Guide", WIRED. [Online]. Available:https://www.wired.com/story/guide-self-driving-cars/. [Accessed: 18- Oct- 2018].
- Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction. MIT press.
- http://webee.technion.ac.il/shimkin/LCS11/ch4_RL1.pdf
- https://pdfs.semanticscholar.org/89b2/6da4e6ac2994dccd0eba0eae9649f442a75c.pdf

Thank You