AWS Deep Racer

Team JCAR

Team JCAR

Ahmed Moustafa

Ramiro Gonzalez

Cristopher Torres

Jaydyn Odor

1

Progress Incentive

- Steps
- 15 steps/sec x 18 sec/lap = 270 steps per lap
- Progress
-

```
# Read input parameters
distance_from_center = params['distance_from_center']
track_width = params['track_width']
speed = params['speed']
waypoints = params['waypoints']
closest_waypoints = params['closest_waypoints']
progress = params['progress']
steps = params['steps']

# Calculate 3 marks that are increasingly farther away from the center line
marker_1 = 0.1 * track_width
marker_2 = 0.25 * track_width
marker_3 = 0.5 * track_width

# initialize thresholds and reward
DIFF_HEADING_THRESHOLD = 6
SPEED_THRESHOLD = 1.8
TOTAL_STEPS = 270      # roughly 15 steps per second, 18 sec default lap
reward = 5
```

```
############################
# Steps and progress, check every 20 steps. The less steps it takes to reach 100% progress means finished faster
############################

# reward less steps with greater progress if faster than 18 sec
if (steps % 20) == 0 and progress/100 > (steps/TOTAL_STEPS):
    reward += progress - (steps/TOTAL_STEPS)*100
```

Waypoints Incentive

```
##############################
# Waypoints: referenced code from https://github.com/MatthewSuntup/DeepRacer/blob/master/reward/reward_final.py
##############################

# finding previous point, next point, and future point
prev_point = waypoints[closest_waypoints[0]]
next_point = waypoints[closest_waypoints[1]]
future_point = waypoints[min(len(waypoints) - 1, closest_waypoints[1]+6)]

# calculate headings to waypoints
heading_current = math.degrees(math.atan2(prev_point[1]-next_point[1], prev_point[0]-next_point[0]))
heading_future = math.degrees(math.atan2(prev_point[1]-future_point[1], prev_point[0]-future_point[0]))

# calculate difference between headings
# check we didn't choose reflex angle
diff_heading = abs(heading_current-heading_future)
if diff_heading > 180:
    diff_heading = 360 - diff_heading

# if diff_heading > than threshold indicates turn
# so when a turn is ahead (high diff_heading)
# penalize high speed, reward slow speed
if (diff_heading > DIFF_HEADING_THRESHOLD) and speed >= SPEED_THRESHOLD:
    reward -= 4
```

Centerline Incentive

```
############################
# Center line incentives
############################

# give higher reward if the car is closer to center line and vice versa
if distance_from_center <= marker_1:
    reward += 5
elif distance_from_center <= marker_2:
    reward += 4
elif distance_from_center <= marker_3:
    reward += 3
else:
    reward -= 4
```

5

Ethical Considerations

- Ensure fairness among users
- Adaptability for different settings
- Reliability and safety precautions
- Inability of moral

Future Bias’s to Consider

- Road conditions / weather bias
- Road obstruction classification
- Civilian recognition bias
- Negative impact on labor

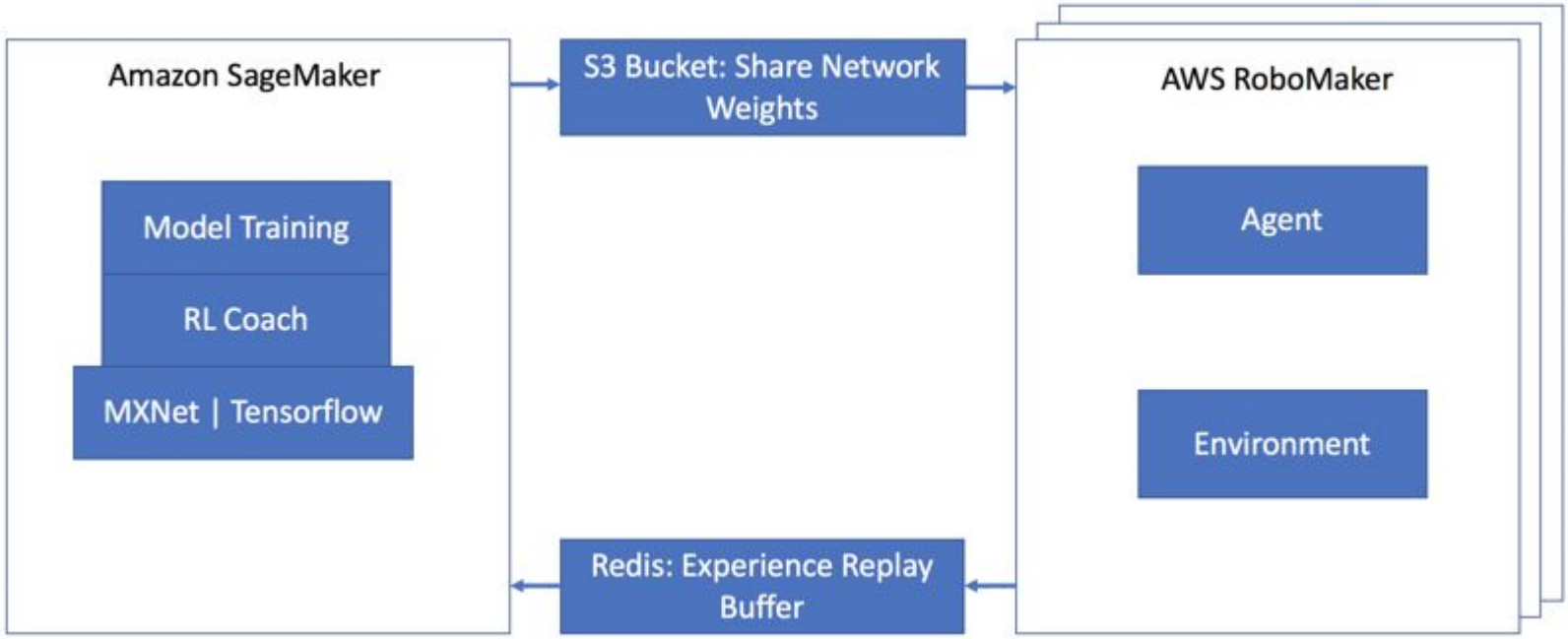
Special Thanks

-

Resources

- https://www.baeldung.com/cs/mdp-value-iteration
- https://towardsdatascience.com/reinforcement-learning-101-e24b50e1d292
- https://docs.aws.amazon.com/deepracer/latest/developerguide/deepracer-how-it-works-action-space.html
- https://docs.aws.amazon.com/deepracer/latest/developerguide/deepracer-how-it-works-solution-workflow.html
- https://github.com/MatthewSuntup/DeepRacer/blob/master/reward/reward_final.py

Questions?


Amazon SageMaker
Model Training
RL Coach
MXNet | Tensorflow
S3 Bucket: Share Network Weights
Redis: Experience Replay Buffer
AWS RoboMaker
Agent
Environment

1. **Agent.** The program you train, with the aim of doing a job you specify.
2. **Environment.** The world, real or virtual, in which the agent performs actions.
3. **Action.** A move made by the agent, which causes a status change in the environment.
4. **Rewards.** The evaluation of an action, which can be positive or negative.