An Exploration of Reinforcement Learning in Complex Environments

Karsten Boettcher

Mentor: Dr. Mario Harper

Utah State University, Computer Science

Introduction

- What is Rocket League?
- Rocket League as an optimization problem
- Reinforcement Learning and Rocket League

Credit: https://steamcommunity.com/sharedfiles/filedetails/?id=900701590

UtahStateUniversity

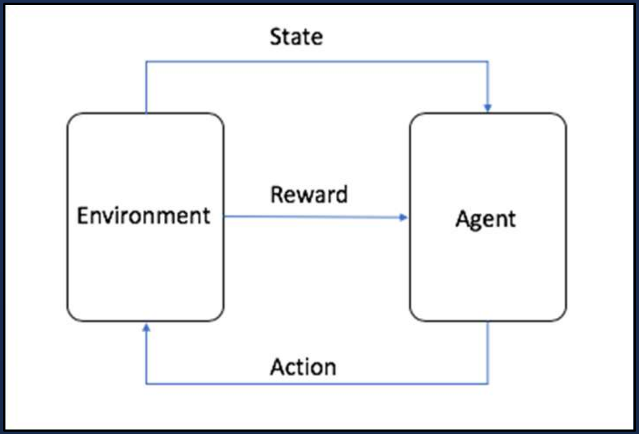
Why Reinforcement Learning?

- Reinforcement Learning aims to:
  - Solve problems beyond human capabilities
  - Test new actions or approaches

UtahStateUniversity

Reinforcement Learning

- Reinforcement Learning training has two components:
  - Agent
  - Environment
- Each timestep, the agent selects an action
- Environment rewards the agent's action
- Environment updates to the new state



Credit: Amazon AWS Machine Learning Blog

UtahStateUniversity

Q-Learning

- Agent learns a q-value
  - Quality of doing *action* in *state*
- Agent selects an action from:
  - *(action, state)* pair with best q-value
  - Random *action* ε-percent of the time
- Agent performs *action*, reward is given

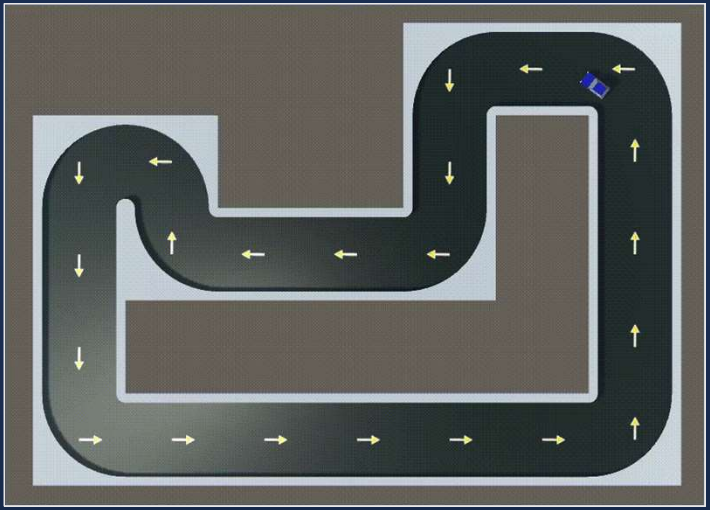


Fig. 1: An agent training to navigate a racetrack as fast as possible.

Q-values are updated from the given reward. Over time the optimal path to the goal is generated.

UtahStateUniversity

Methodology

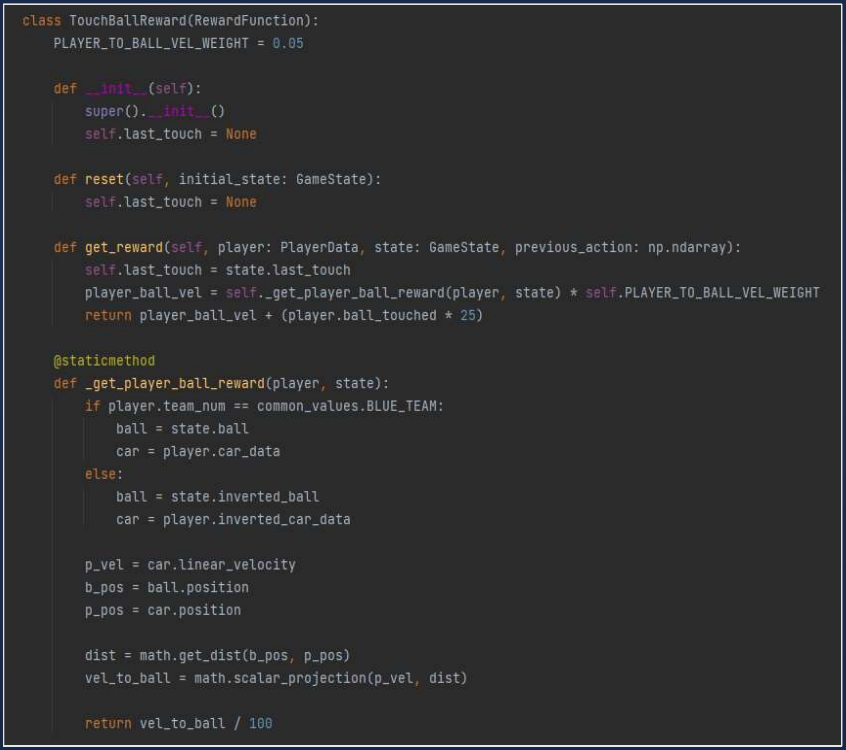
Credit: https://www.usgamer.net/articles/rocket-league-xbox-one-review

- Determine the objective goal:
  1. Maintain the fastest velocity possible
     - Training episode ends after 5 minutes
  2. Hit the ball efficiently
     - Episode ends once ball is hit
     - Episode ends after 5 minutes

UtahStateUniversity

Methodology

- Define rewards:
  1. For moving as quickly as possible:
     - Reward the bot its instantaneous speed
  2. For hitting the ball:
     - Reward the bot its velocity to ball
     - Large reward for touching the ball



```python
class TouchBallReward(RewardFunction):
    PLAYER_TO_BALL_VEL_WEIGHT = 0.05

    def __init__(self):
        super().__init__()
        self.last_touch = None

    def reset(self, initial_state: GameState):
        self.last_touch = None

    def get_reward(self, player: PlayerData, state: GameState, previous_action: np.ndarray):
        self.last_touch = state.last_touch
        player_ball_vel = self._get_player_ball_reward(player, state) * self.PLAYER_TO_BALL_VEL_WEIGHT
        return player_ball_vel + (player.ball_touched * 25)

    @staticmethod
    def _get_player_ball_reward(player, state):
        if player.team_num == common_values.BLUE_TEAM:
            ball = state.ball
            car = player.car_data
        else:
            ball = state.inverted_ball
            car = player.inverted_car_data

        p_vel = car.linear_velocity
        b_pos = ball.position
        p_pos = car.position

        dist = math.get_dist(b_pos, p_pos)
        vel_to_ball = math.scalar_projection(p_vel, dist)

        return vel_to_ball / 100
```

Fig. 2: TouchBallReward code which defines how the agent is rewarded.

UtahStateUniversity

Fig. 3: Rocket League agent during early training. (0 – 100,000 timesteps)

UtahStateUniversity

Results: Max Speed

50:00
BALL CAM
PRESS SPACEBAR TO TOGGLE

Fig. 4.1: Rocket League agent training for maximum speed, 5 million timesteps.

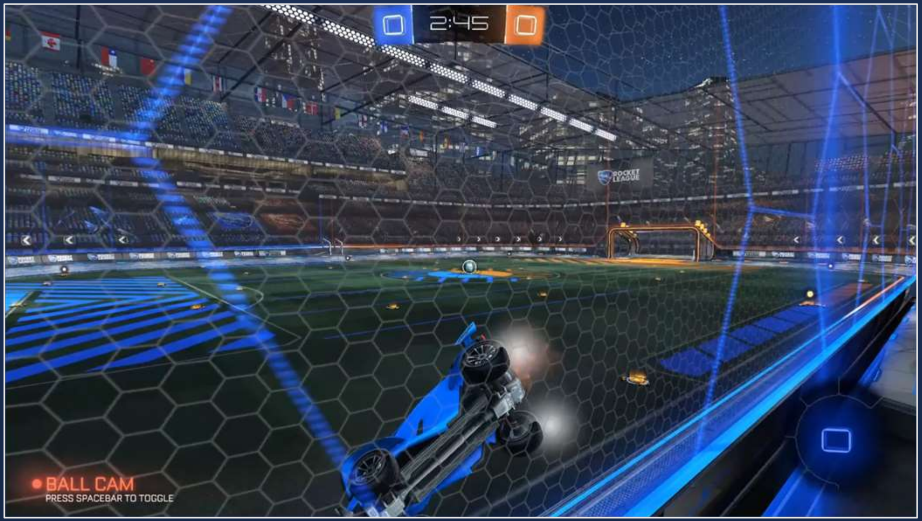


Fig. 4.2: Rocket League agent training for maximum speed, 15.5 million timesteps.

UtahStateUniversity

Results: Hitting the Ball

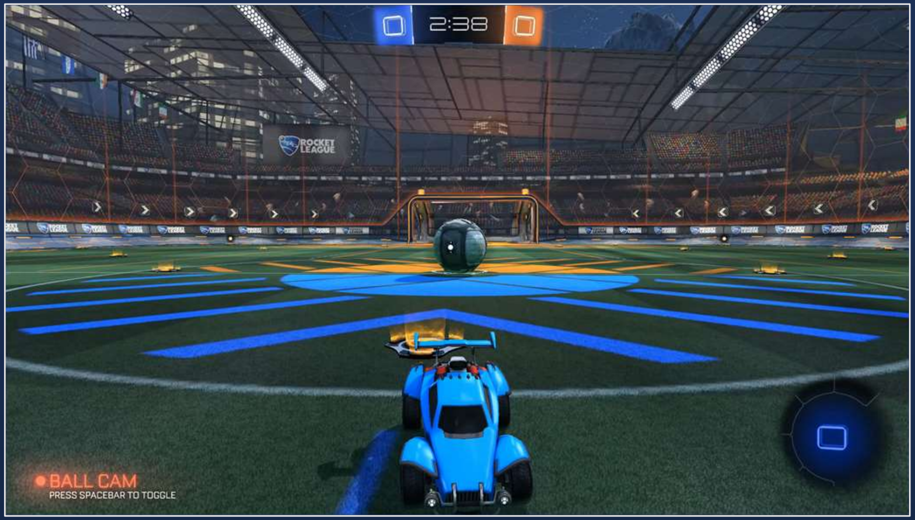


Fig. 5.1: Rocket League agent training to hit the ball, 500,000 timesteps.

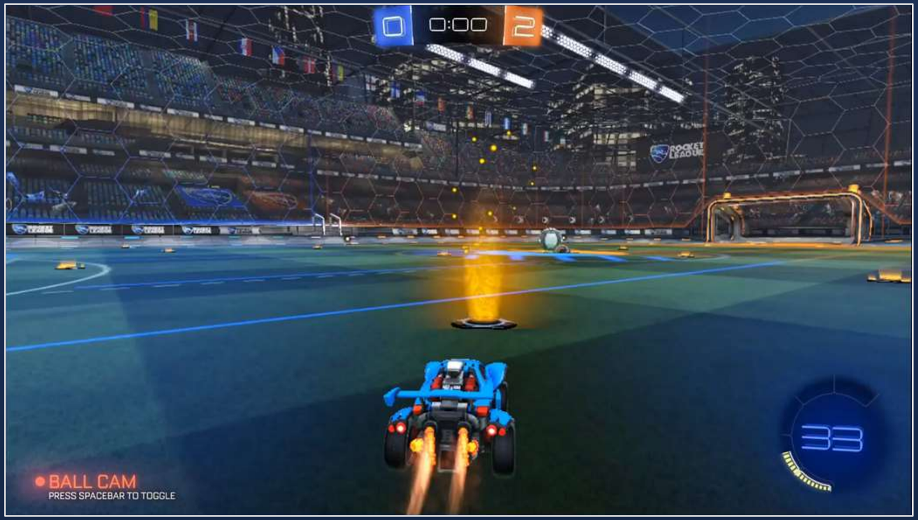


Fig. 5.2: Rocket League agent training to hit the ball, 3 million timesteps.

UtahStateUniversity

Results

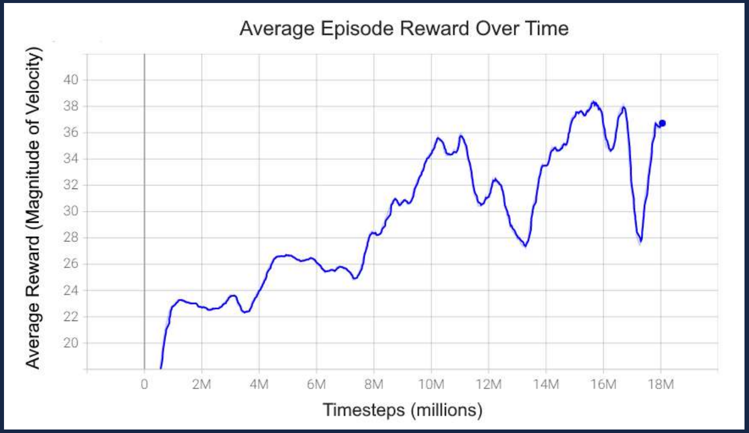


Fig. 6: Average episode reward over time for objective 1. (Maintain the fastest velocity possible)

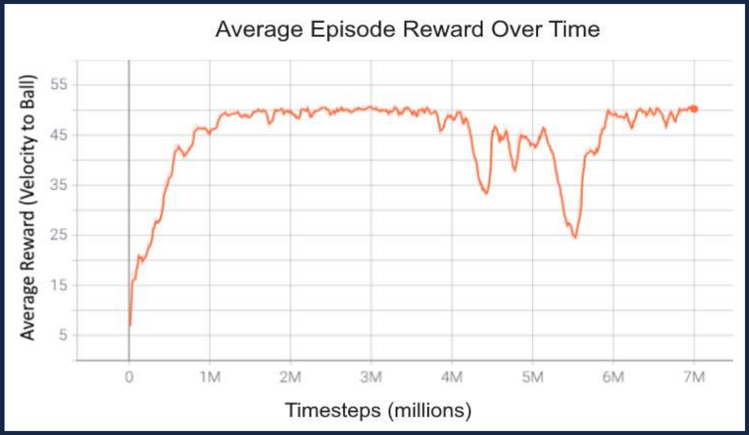


Fig. 7.1: Average episode reward over time for objective 2. (Hitting the ball)

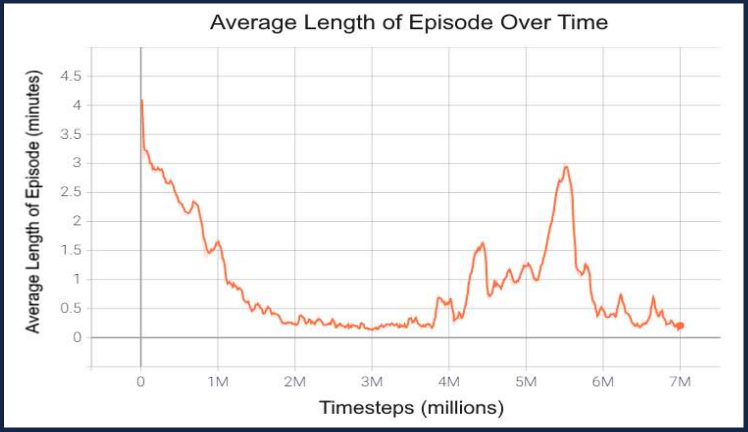


Fig. 7.2: Average length of episode over time for objective 2. (Hitting the ball)

UtahStateUniversity

Conclusion

The Rocket League bot learned to execute simple tasks efficiently.

Our research about agents learning in complex environments shows promise for more difficult tasks.

UtahStateUniversity

Contact

Karsten Boettcher
Utah State University
karstenbruce@gmail.com

Dr. Mario Harper
Utah State University
mario.harper@usu.edu

UtahStateUniversity

For More Information:

- https://gym.openai.com/
- https://rlbot.org/
- https://github.com/lucas-emery/rocket-league-gym