

AWS DEEP RACER

Reinforcement Learning – CSE 546

Dec 08, 2020

Team Members:

Upmanyu Tyagi (50289812)

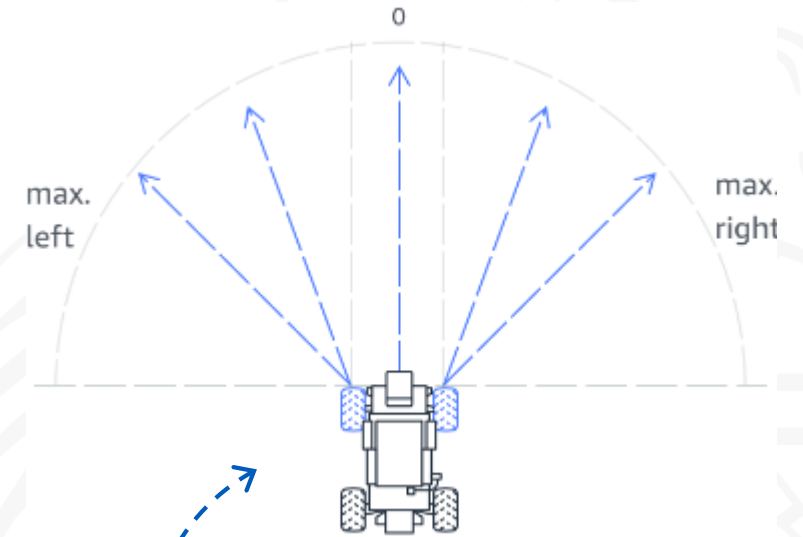


Project Description

- Amazon defines AWS DeepRacer as “an autonomous 1/18th scale race car designed to test RL models by racing on a physical track.”
- It will be able to make short termed decisions while aiming to optimize for a longer term goal.
- Action space is defined by a combination of few features.
- Speed, Steering Angle, Torque(wheels rotating forward or reverse)



AWS DeepRacer



Action Space

BACKGROUND



Proximal Policy Optimization (PPO) Algorithm

- Proximal Policy Optimization (PPO) algorithm has been used. It is a derivative of the policy gradient method.
- Policy gradient method works by computing an estimator of the policy gradient and plugging it into a stochastic gradient ascent algorithm.
- It uses 2 neural networks namely a policy network and a value network.
- The policy network takes image as an input and decides which actions are to be taken.
- The value network will calculate the cumulative reward which is expected if the image is given as an input.
- The policy network will interact with the simulator.
- If it is to be deployed in the real-world, policy network is deployed.

Amazon Web Services (AWS)

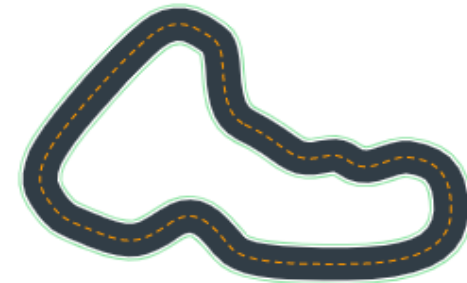
- As we are running in simulation environment we are using AWS SageMaker.
- AWS (Amazon Web Services) is an evolving cloud computing platform provided by Amazon which is comprehensive as well.
- AWS has many types tools and solutions for software developers and companies which can easily be used in data centers in around 190 countries.
- AWS SageMaker is a machine learning platform that is cloud based. It was launched in November 2017.

IMPLEMENTATION



Tracks

- There are various types of racing tracks for AWS DeepRacer that are available namely, re:Invent 2018, Baadal Track and Fumiaki Track.
- Re:Invent being the basic type of track our first models are tested on re:Invent 2018.
- Eventually we tested few models on Fumiaki Track.
- We chose Baadal Track for our final model to be trained on.



- **Baadal Track**
Baadal is the Hindi word for cloud. The Baadal track combines long arching straightaways perfect for passing opportunities coupled with tight windings corners.


Length: 39 m (128')
Width: 107 cm (42")


Race Types:


- There are three types of Races that we can choose to train DeepRacer on, (time trial, object avoidance and head-to-head racing) and participate in the competitions.
- Time trial is focused on finishing the lap in fastest time.
- Object avoidance is trained to avoid the obstacles in the track without colliding with them.
- Head-To-Head is to compete with the BOT races racing in the track along with our CAR.

Race type

Choose a race type

Time trial
The agent races against the clock on a well-marked track without stationary obstacles or moving competitors.


Object avoidance
The vehicle races on a two-lane track with a fixed number of stationary obstacles placed along the track.


Head-to-head racing
The vehicle races against other moving vehicles on a two-lane track.


Implementation

- 8 models were tested for AWS DeepRacer implementation.
- PPO algorithm was used on AWS Console.
- The maximum turning angle was taken as 30 degrees.
- The granularity was taken as 5.
- We worked on the implementation of time trial racing on AWS DeepRacer.
- One model was trained for Head-to-head
- Gradient descent batch size is 64.
- Entropy is 0.01.
- Discount factor taken is 0.999.
- Loss type is Huber.
- Learning rate is 0.0002.
- Number of experience episodes between each policy-updating iteration are 10.
- Number of epochs are 6.

RESULTS



Evaluation Results

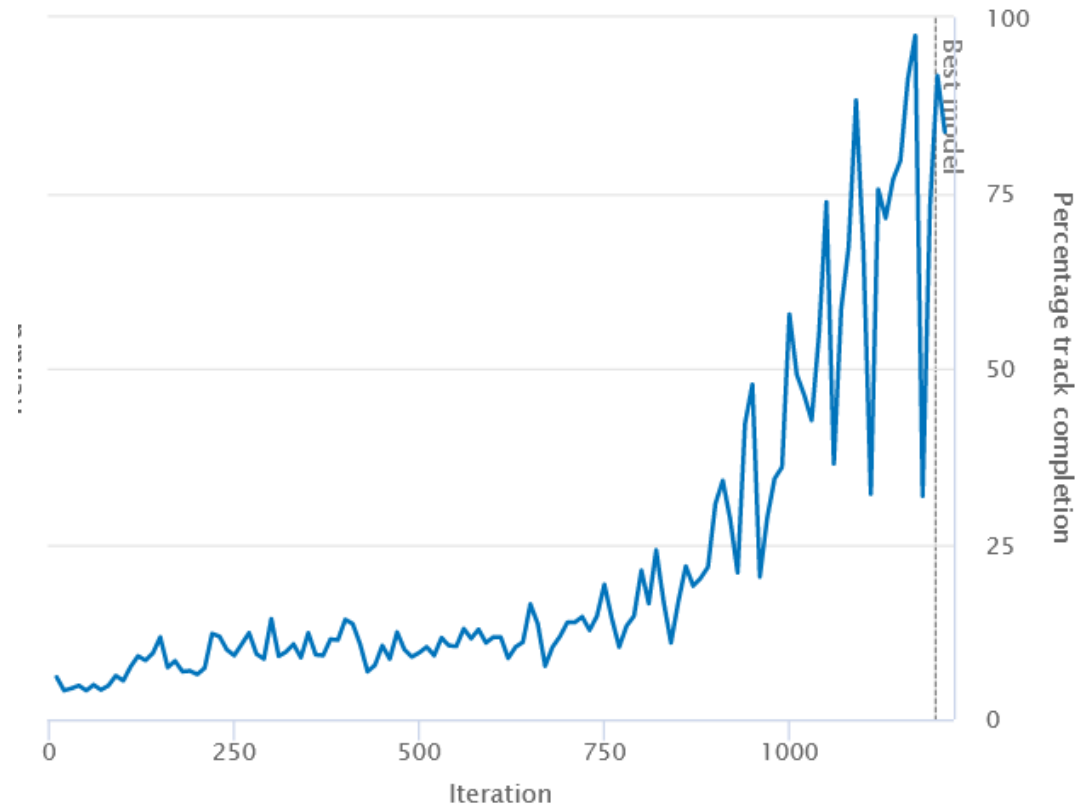
- The track used for training is “Baadal Track”.
- After training is done the model is evaluated on the same track over 5 trials.
- The lap finish time for each trial is seen in the results.
- This figure shows the trial result in percentage track completed and the corresponding status including the time taken to complete the whole track by the agent.

Evaluation results

Trial	Time	Trial results (% track completed)	Status
1	00:00:27.214	100%	Lap complete
2	00:00:27.590	100%	Lap complete
3	00:00:27.256	100%	Lap complete
4	00:00:27.733	100%	Lap complete
5	00:00:27.662	100%	Lap complete

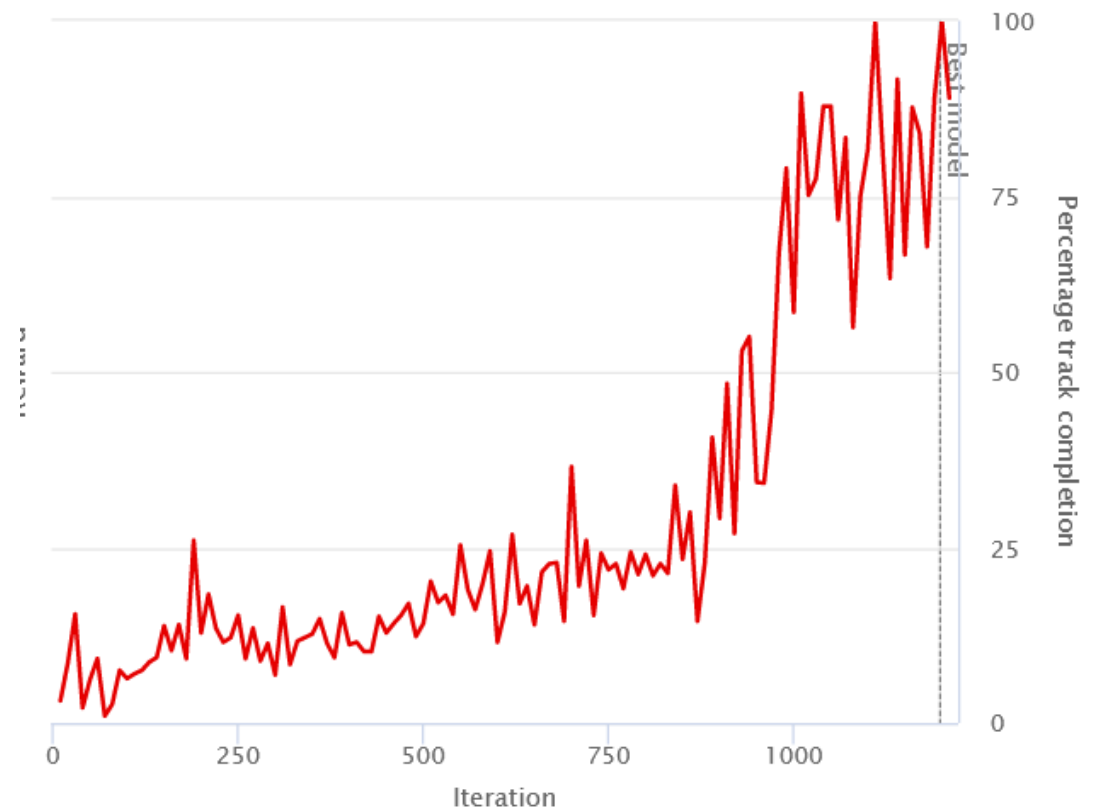
Completion percentage training

- It is the average percentage completion per episode while training.
- The figure shows the average completion rate per lap during training phase.
- The model approximately converges after 1000 iterations.



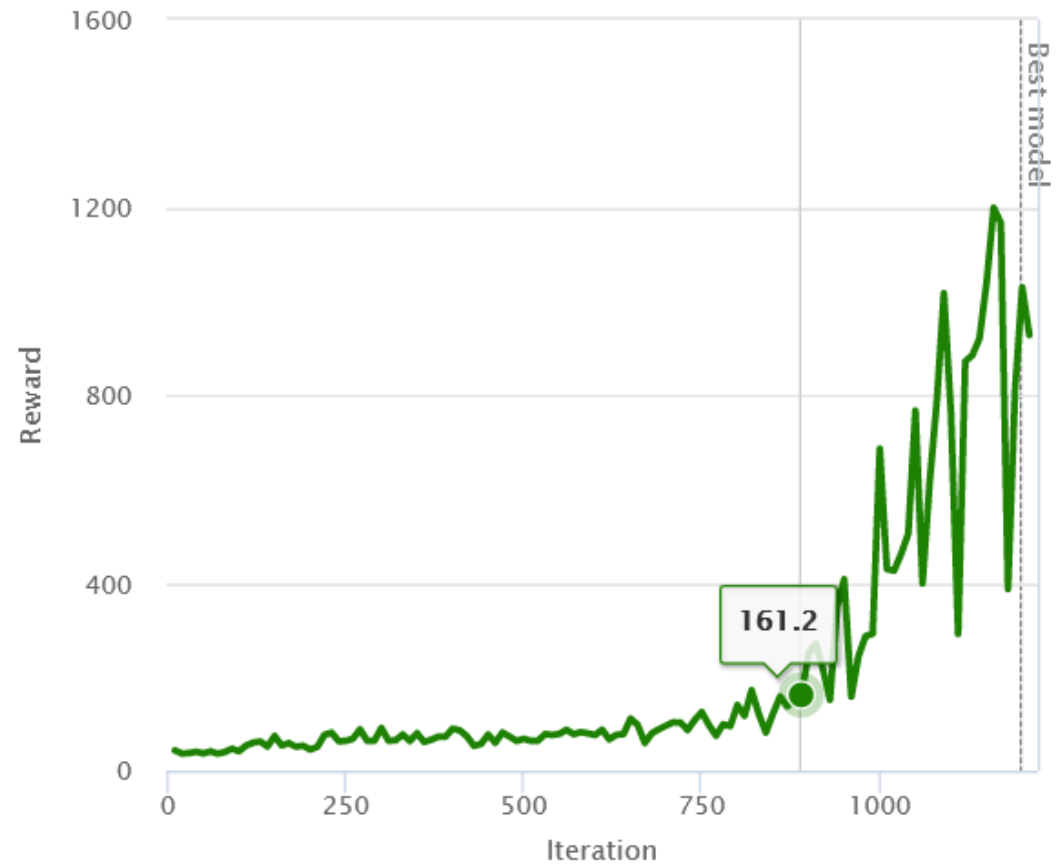
Evaluation of completion percentage

- It is the average percentage completion per episode while evaluating.
- The figure shows the average completion rate of the lap during evaluation.
- The average evaluation per lap increase with iterations as the model gets better.



Average Reward

- It is the average reward plot over time while training per each episode.
- The figure shows the average reward received by the agent per lap.
- The average reward significantly increases after the 1000 iterations when the model converges.
- As the reward function is defines as the agent receives higher reward for finishing the lap.



KEY OBSERVATIONS



Key Observations

- If the car is encouraged to go faster, then it often leaves the track.
- If the car is encouraged to stay in the center of the track, then it goes slow.
- A balance needs to be found between the speed of the car and its capability to stay on track.
- We got the best results when the reward function is well rewarding when at least 2 wheels of the car are on track.
- If learning rate is small then it takes more time for the model to converge. If it is too big, then it may not converge properly or sometimes it suffers from overfitting as it did not perform well on other tracks.

THANK YOU

