AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

DPR301-R2

SPONSORED BY INTEL

Use machine learning analysis tools with AWS DeepRacer

Don Barber
Sr. Manager, Solutions Architects, Strategic Accounts
AWS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Your journey so far

Taken the DPR201 Get rolling with machine learning on AWS DeepRacer session

Trained and evaluated an AWS DeepRacer model

Submitted a model to the leaderboard



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Provision the workshop environment

Take your reward functions to the next level

How to interpret training graphs

Analyzing training logs and models for insight

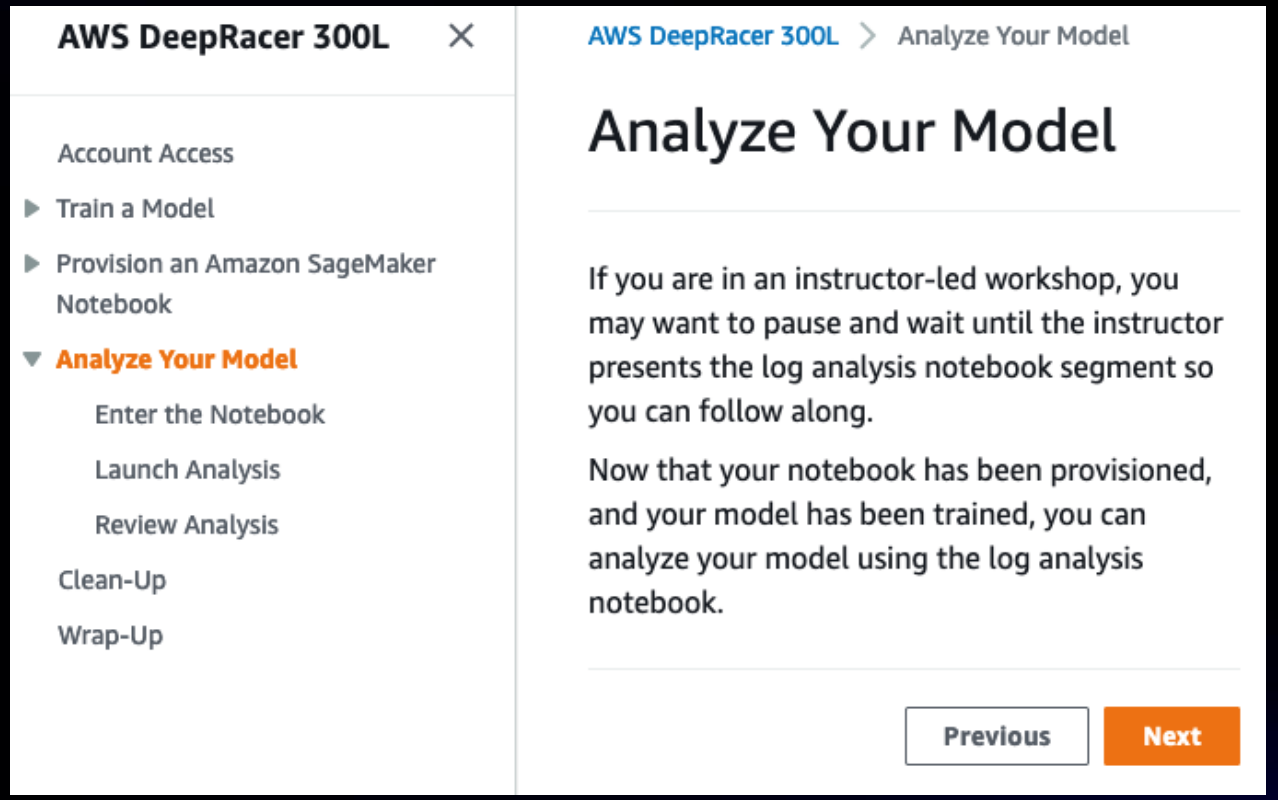
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Provision the workshop environment

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Visit https://deepracer300.workshops.aws.dev/

Proceed until the ‘Analyze Your Model’ section



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Under the hood

- 1:18 4WD scale car
- Intel Atom processor
- Intel distribution of OpenVINO toolkit
- Stereo camera (4MP)
- 360 Degree 12 Meters scanning radius LIDAR sensor
- System memory: 4 GB RAM
- 802.11ac Wi-Fi
- Ubuntu 16.04.3 LTS
- ROS Kinetic



OpenVINO™

intel. ATOM

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Reward function tricks

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Reminder of a reward function

```
def reward_function(params):

    track_width = params['track_width']
    distance_from_center = params['distance_from_center’]

    in_center = 0.25 * track_width

    if distance_from_center <= in_center:
        reward = 1
    else:
        reward = 1e-3  # not in center

    return reward
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Reward function parameters

```
all_wheels_on_track
x
y
closest_objects
closest_waypoints
distance_from_center
is_crashed
is_left_of_center
is_offtrack
is_reversed
heading
objects_distance
objects_heading
objects_left_of_center
objects_location
objects_speed
progress
speed
steering_angle
steps
track_length
track_width
waypoints
```

List available at: https://docs.aws.amazon.com/deepracer/latest/developerguide/deepracer-reward-function-input.html

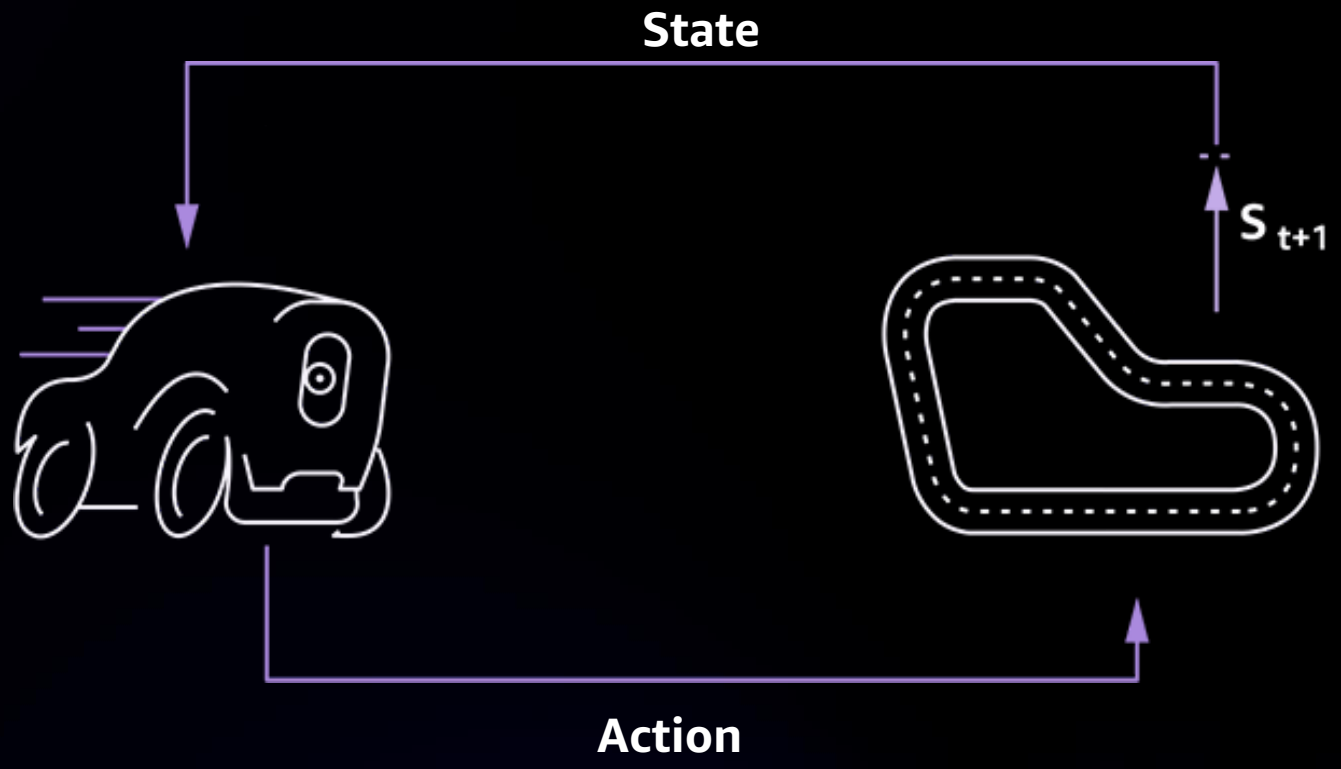
aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Useful input parameter: Steps

Number of steps completed; will be reset when off track

Easy way to improve the finish time



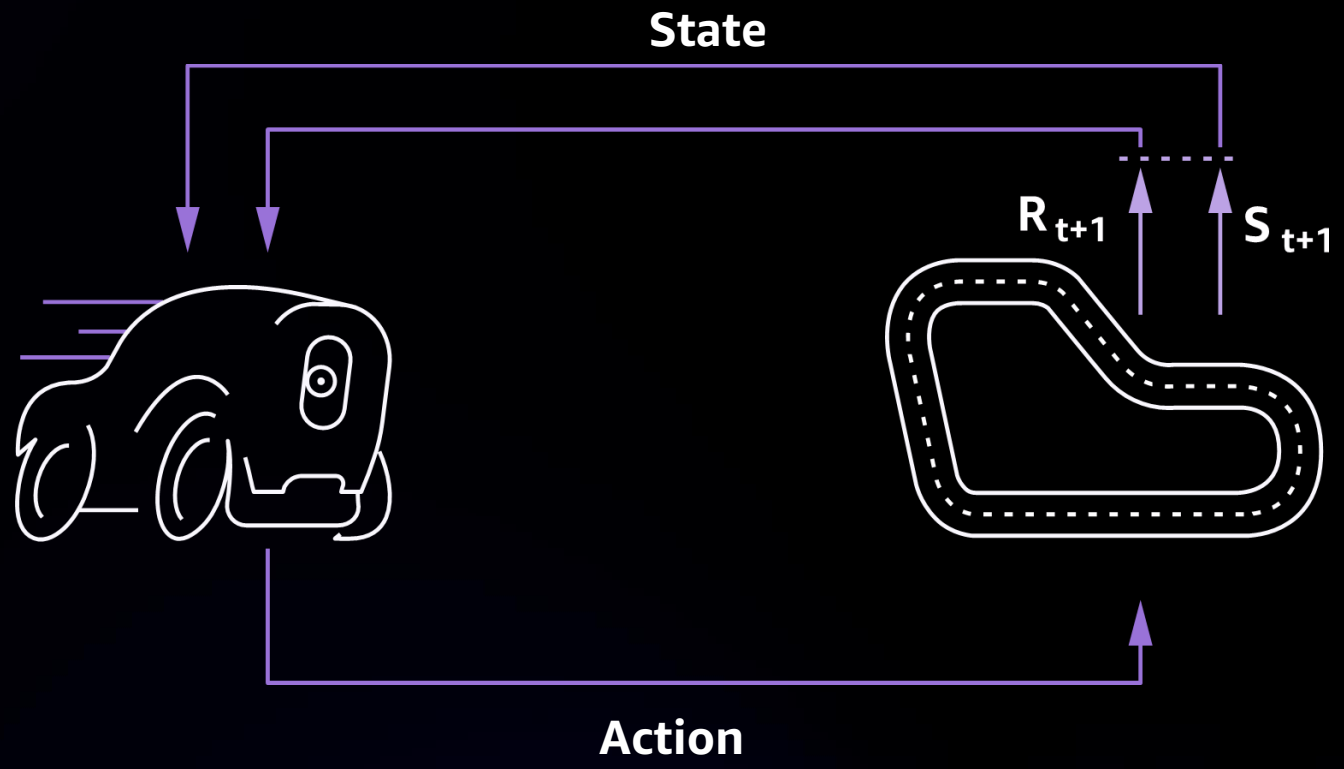
aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Useful input parameter: Steps

Number of steps completed; will be reset when off track

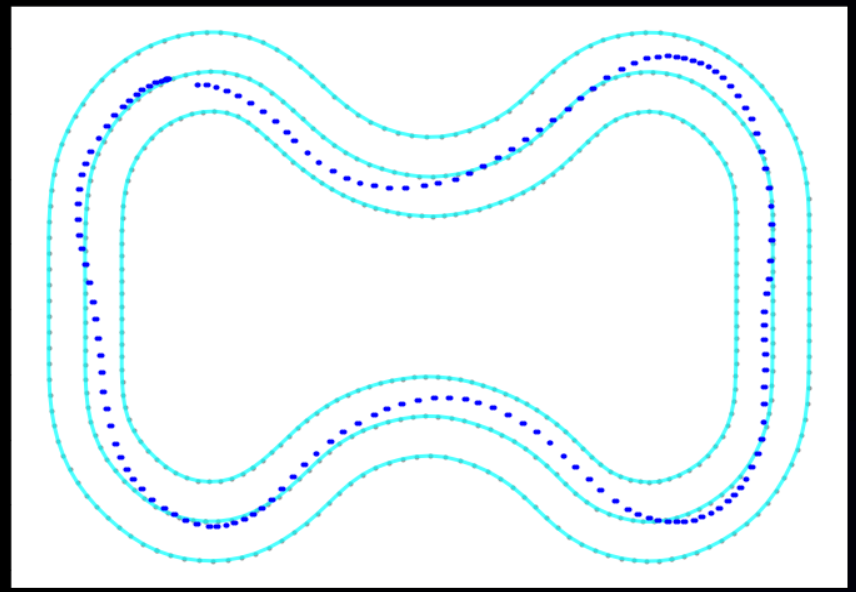
Easy way to improve the finish time

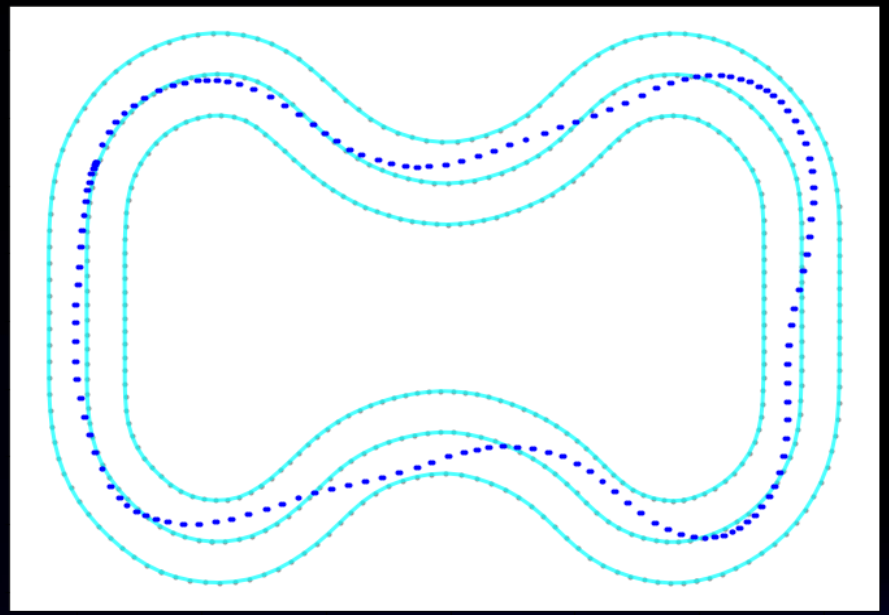


© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Comparison and sample code

```
16
17      if steps > 0:
18          reward = ((progress*150)/steps)**2
19      else:
20          reward= 1
21
```

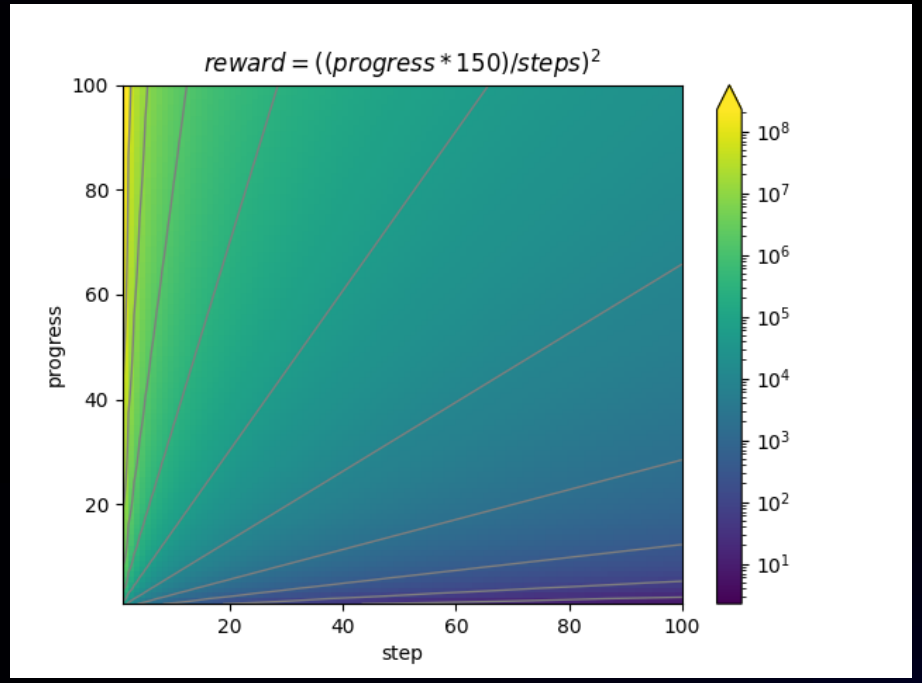




© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Visualize your reward function

```
16
17      if steps > 0:
18          reward = ((progress*150)/steps)**2
19      else:
20          reward= 1
21
```

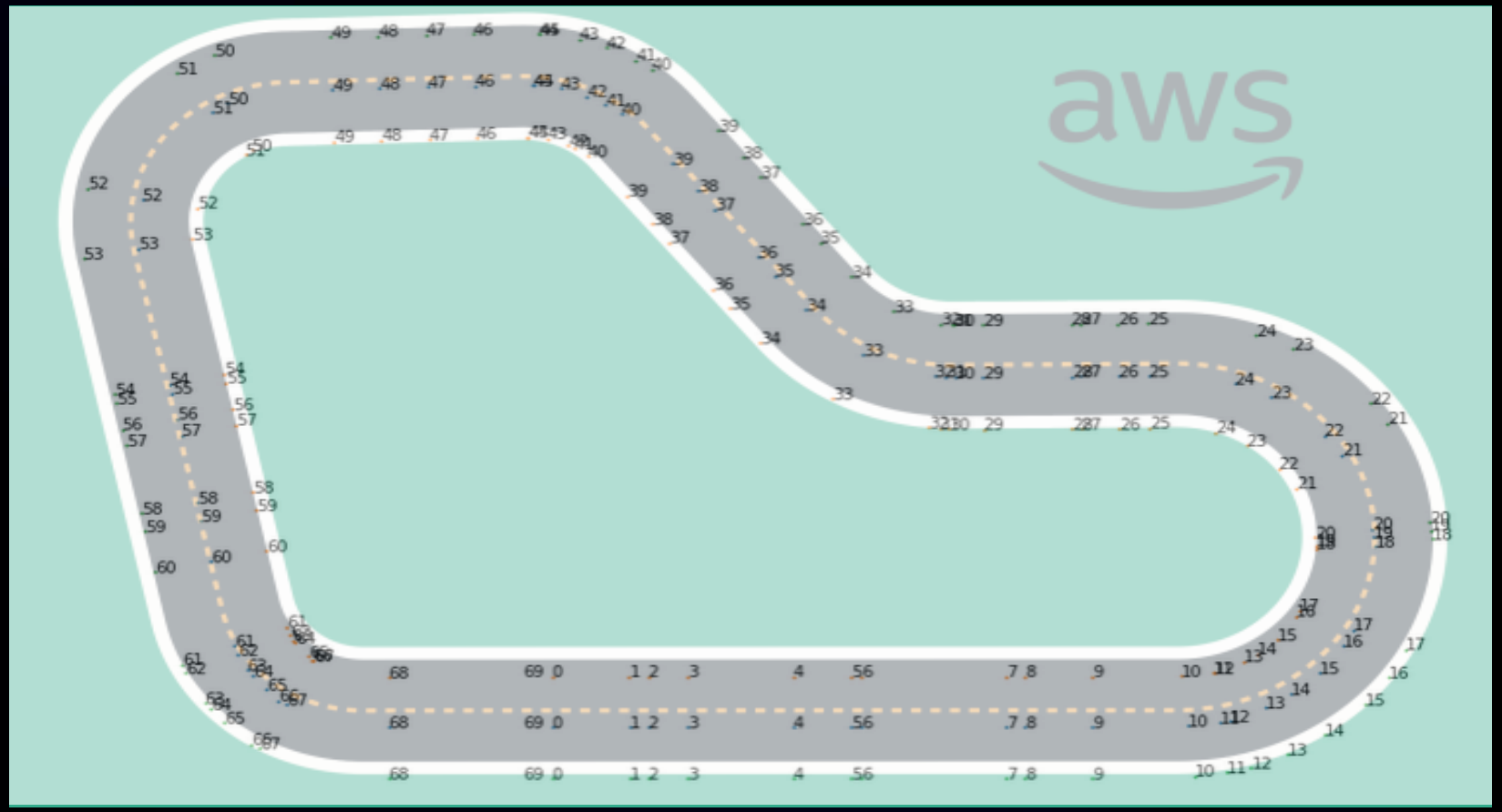


© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Useful input parameter: Waypoints

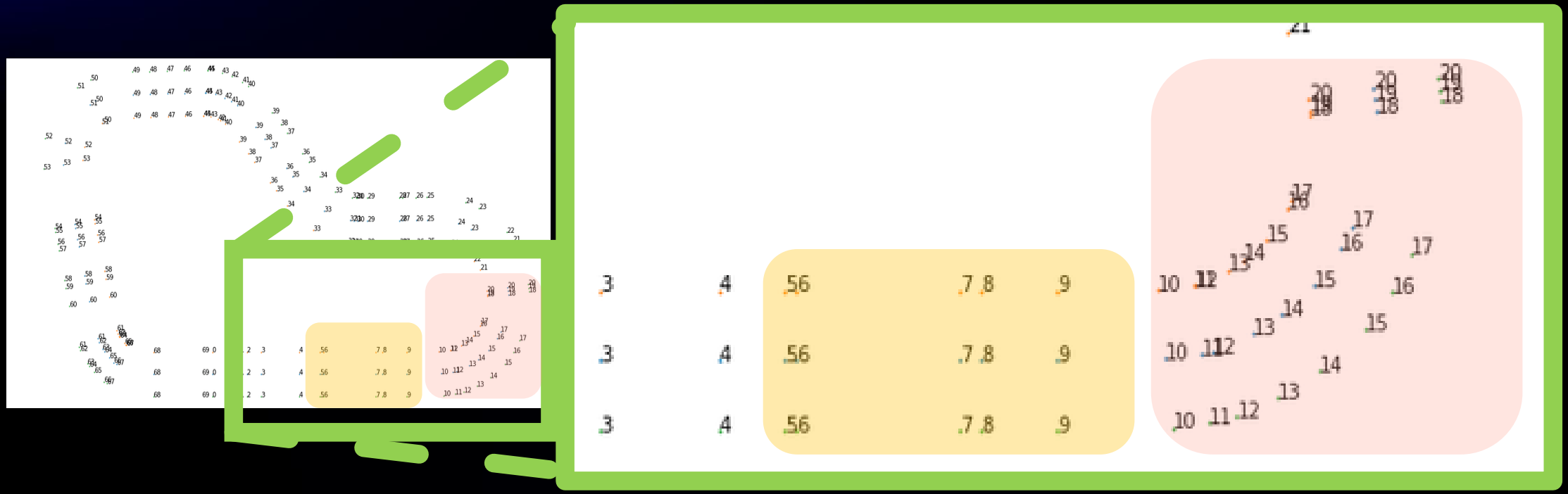
An ordered list of track-dependent milestones along the track center

Lets you set different criteria at different positions



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Using waypoints to reward different behavior



```
21
22      R = [5,6,7,8,9]
23      L = [10,11,12,13,14,15,16,17,18,19,20]
24
25      if (closest_waypoints[0] in (L)) and (is_left_of_center == True):
26          reward = 1
27      elif (closest_waypoints[0] in (R)) and (is_left_of_center == False):
28          reward = 1
29      else:
30          reward = 0.3
31
32
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Use Python modules

Consider using Python built-in modules such as numpy, scipy, and shapely to reduce the heavy lifting of waypoint math:

```
from shapely.geometry import Point, Polygon
from shapely.geometry.polygon import LinearRing, LineString

track = LinearRing(params['waypoints'])

first_object_location = Point(params['objects_location'][0])
object_distance_from_center_line = first_object_location.distance(track)
```

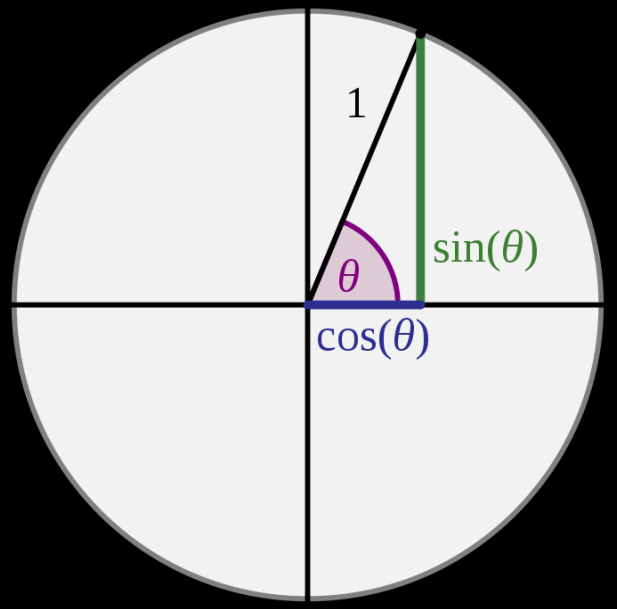
aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Geometry and trigonometry

Distance to waypoint: Use Pythagorean Theorem

```
Distance = math.sqrt((w[0]-x)**2+(w[1]-y)**2)
```



Angle to waypoint: Use atan2

```
angle = math.atan2(w[1]-y,w[0]-x)
```

Answer is in radians, so multiply by 180/π before comparing

Be aware of when angles move from +180 to -179.9

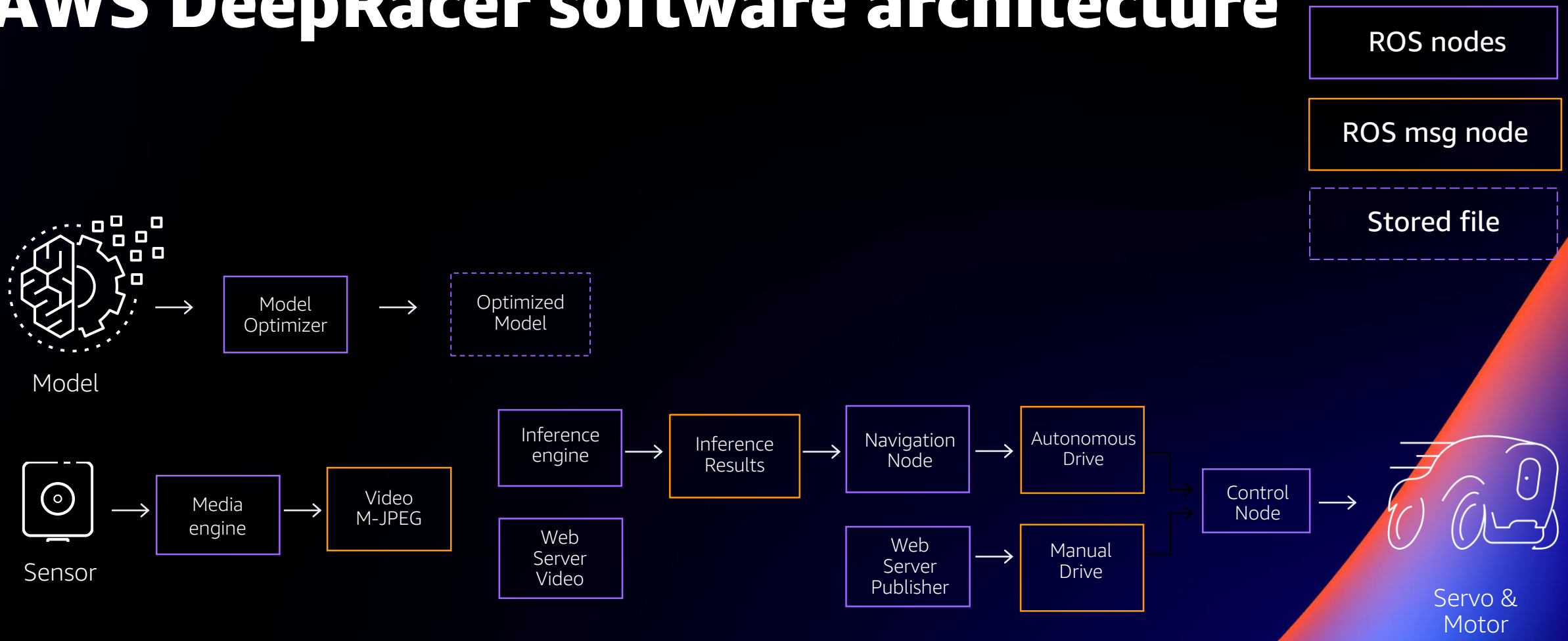
aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Training observation

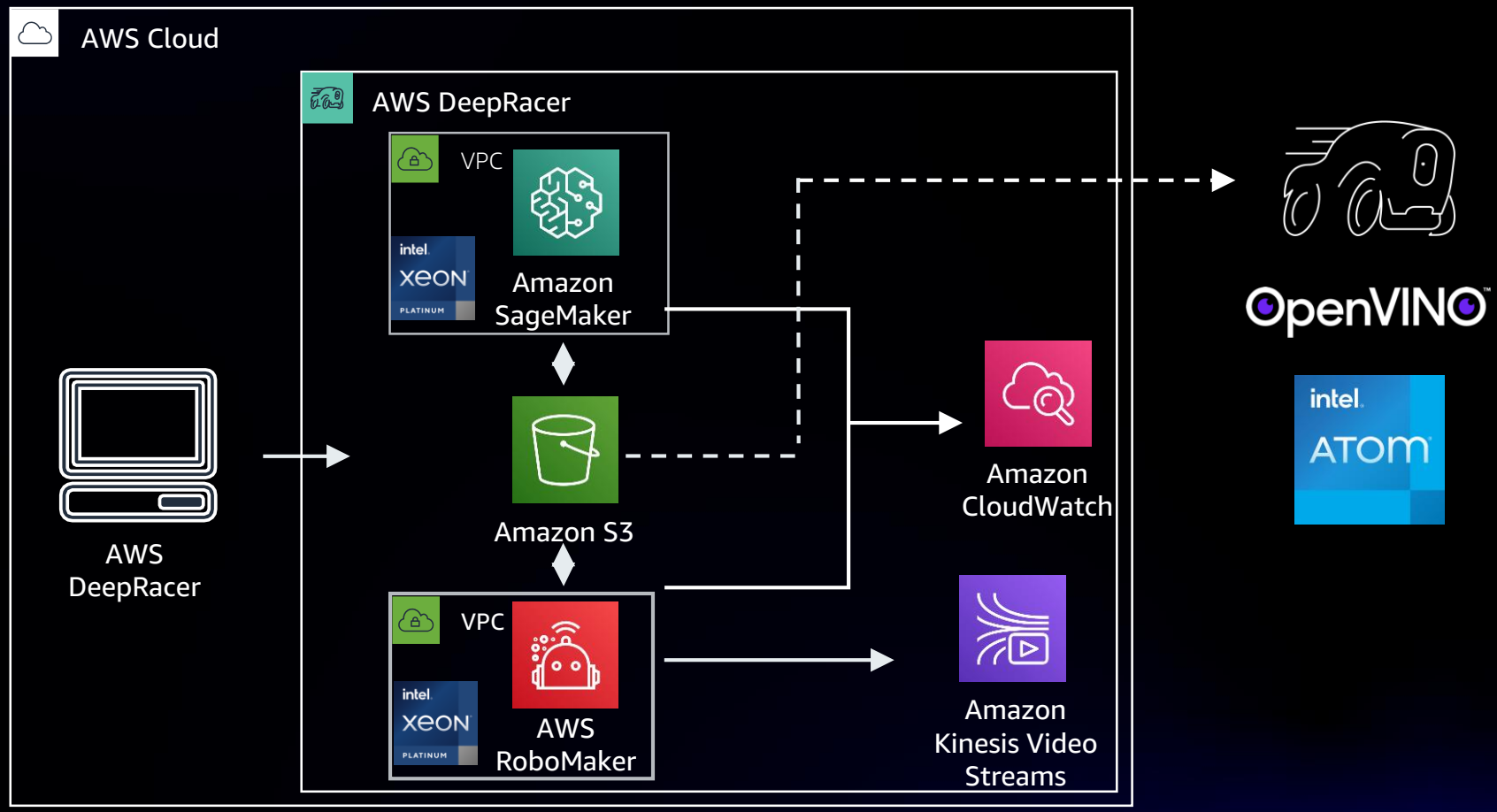
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS DeepRacer software architecture



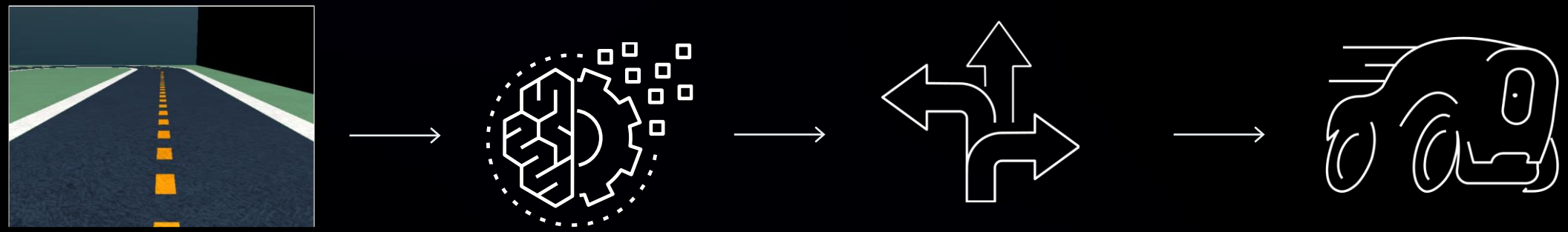
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS DeepRacer simulator architecture



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Optimizing and inferencing with OpenVINO



Input data

OpenVINO™ optimized model

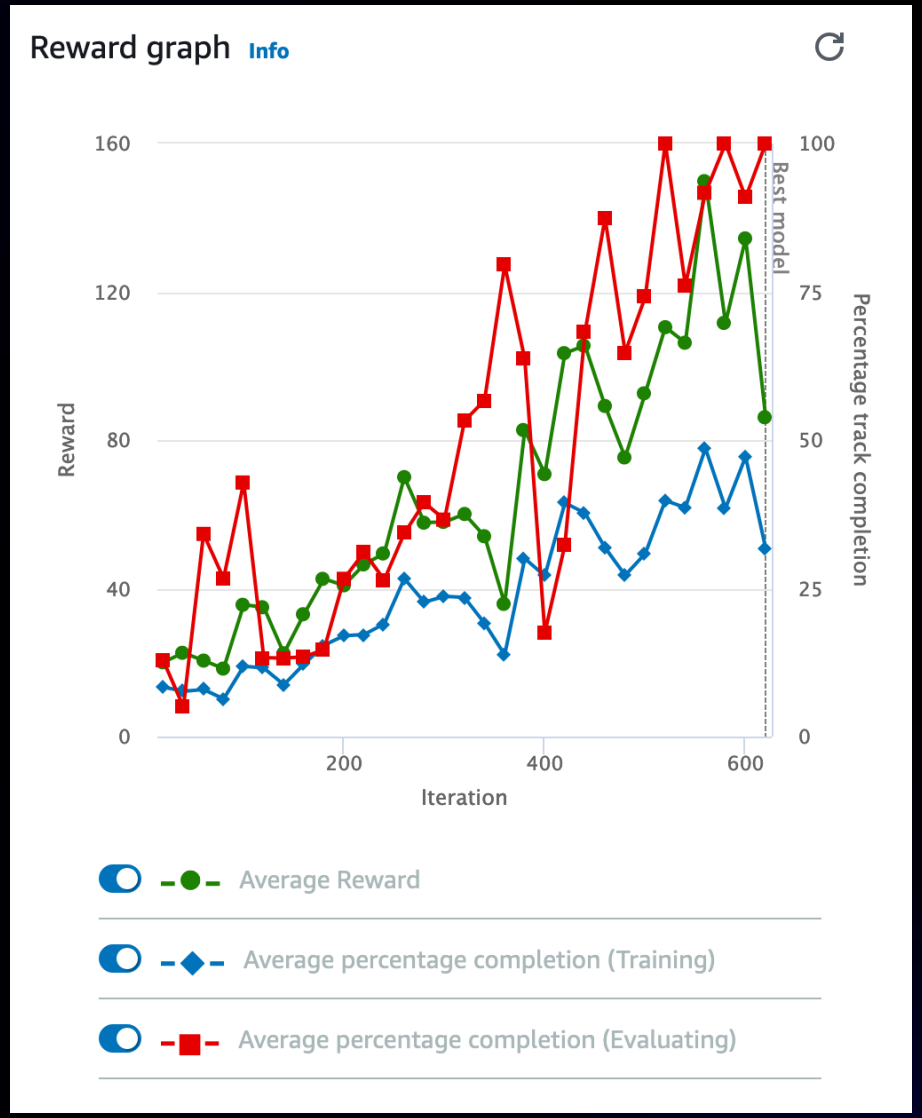
OpenVINO™ inference results

Racing

Free Download: software.intel.com/openvino-toolkit
Open Source version: 01.org/openvinotoolkit

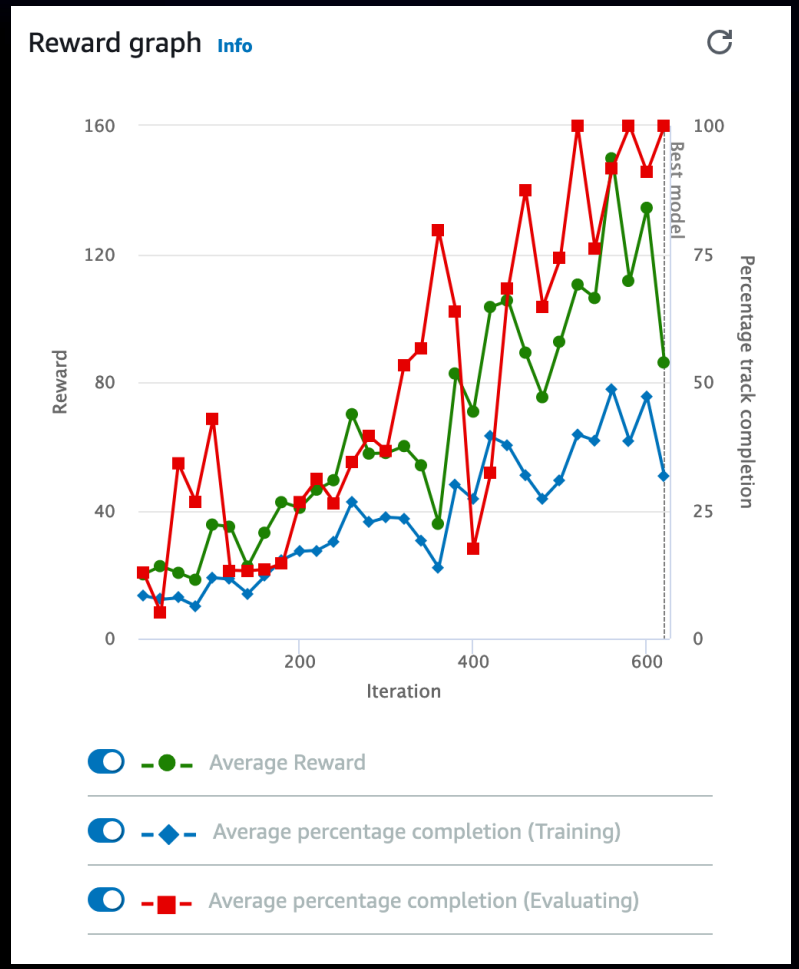
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Do you need more training time?



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why are the training and evaluation lines separate?



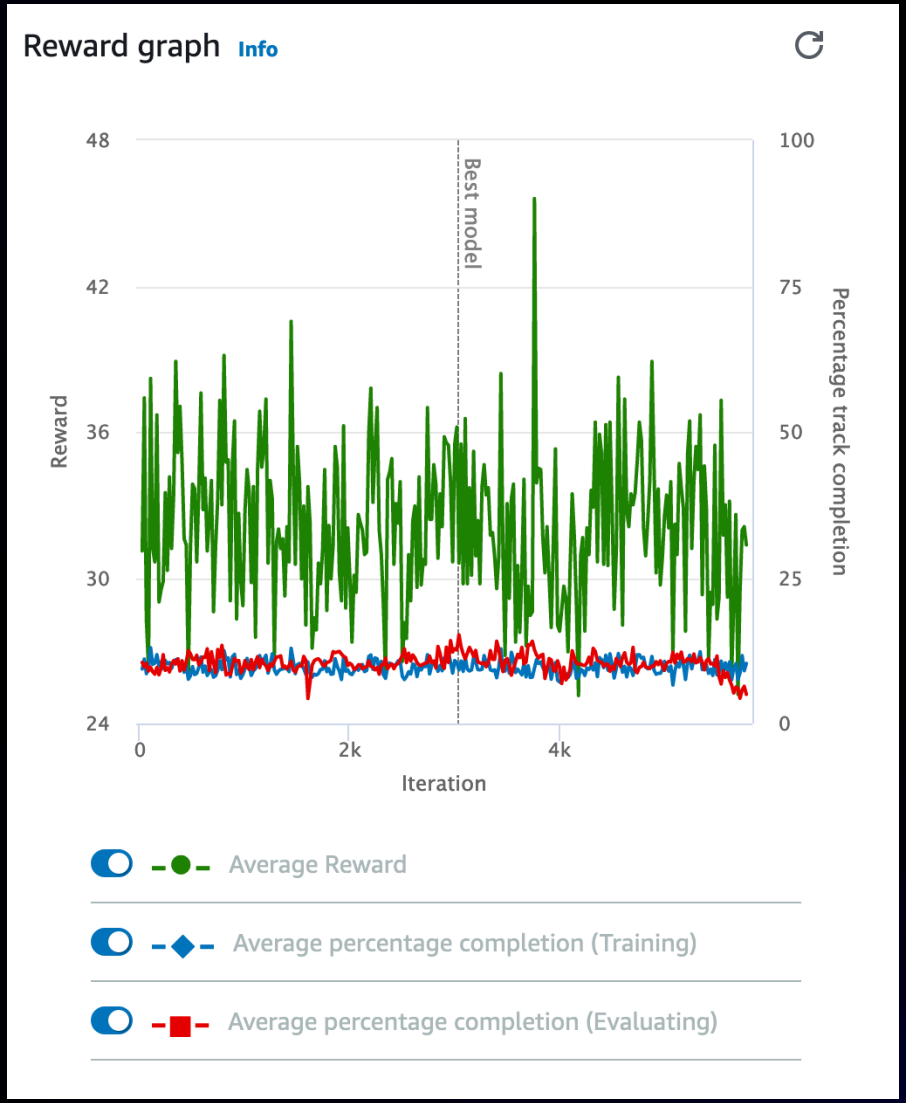
The training graph is the performance of the vehicle while experience is being gathered

After this experience is gathered, the neural network is updated in the background

While the model is being updated, the existing model is evaluated; the evaluation graph is the performance of the vehicle

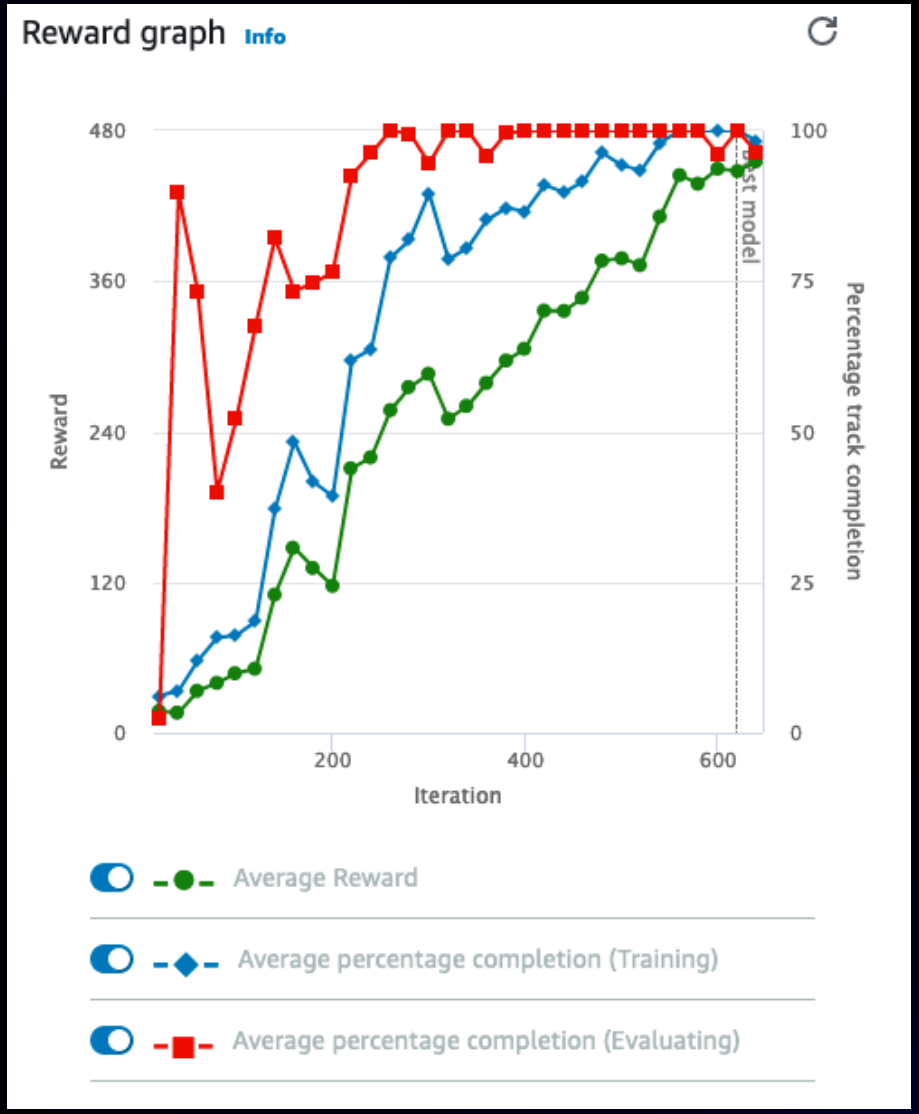
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Is training not improving?



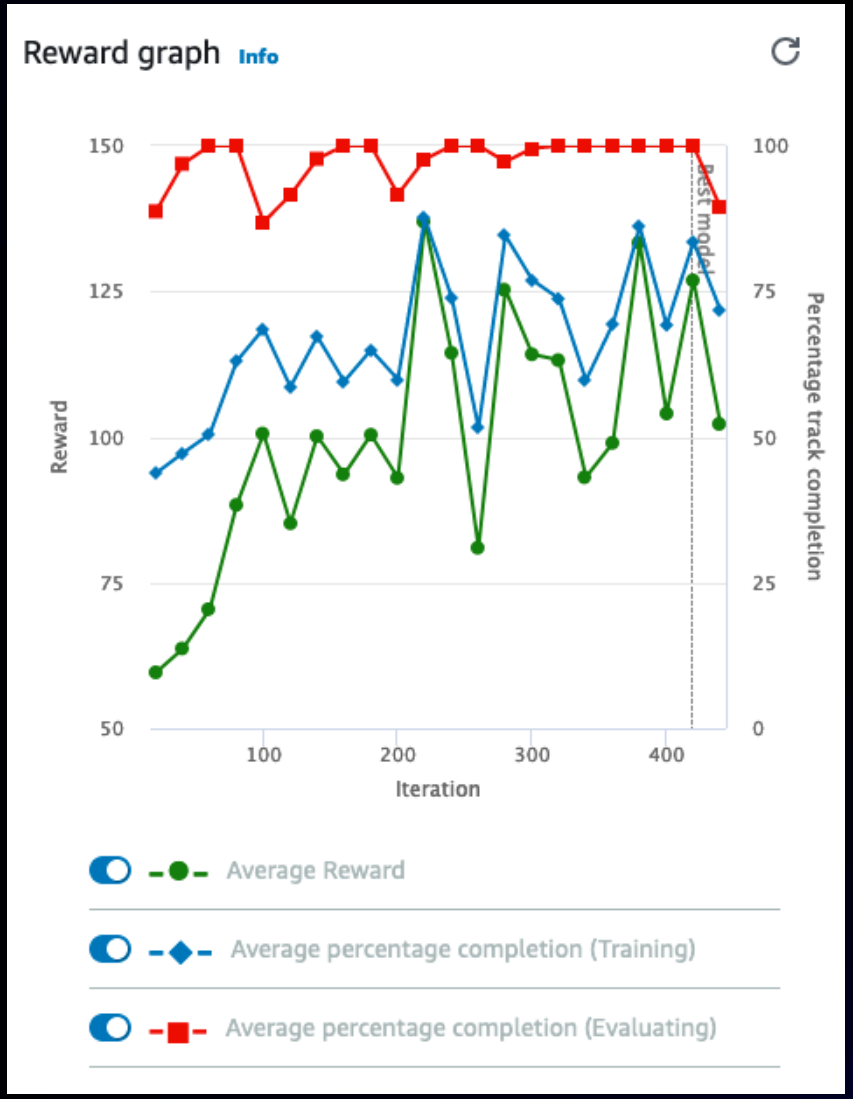
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Are you overfitting?



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

More training with adjusted hyperparameters?



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Analyzing logs and models

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Example log

Fields:

```
episode, step, x-coordinate, y-coordinate, heading,
steering_angle, speed, action_taken, reward, job_completed,
all_wheels_on_track, progress,  closest_waypoint_index,
track_length, time.time(), episode_status
```

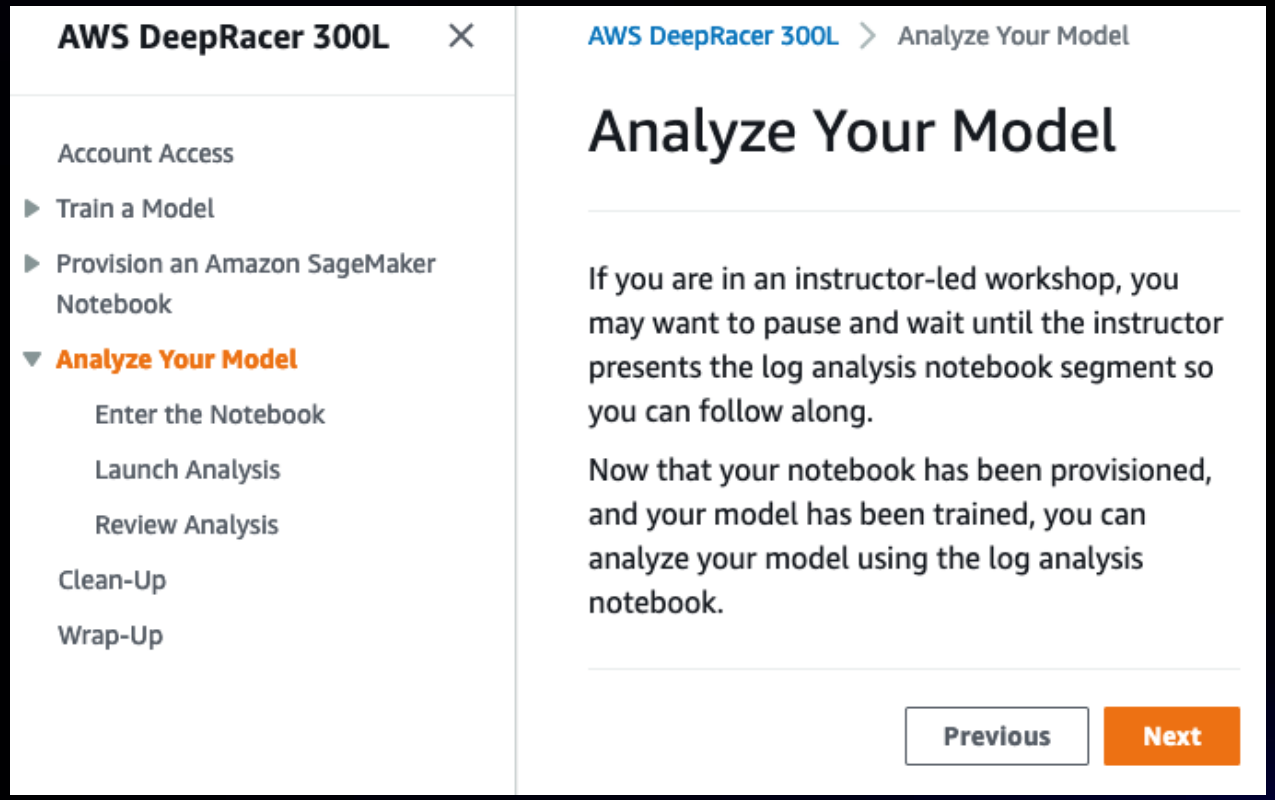
Sample entry:

```
619,44,2.9598,1.0960,29.5347,30.00,0.33,6,0.1333,False,True,10
.0160,8,17.43,1585773790.476588,in_progress
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Log analysis notebook

Proceed with the ‘Analyze Your Model’ section



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Visualize the track and waypoints

GOOD FOR DEVELOPING REWARD FUNCTIONS AND ANALYZING BEHAVIOR BY WAYPOINT



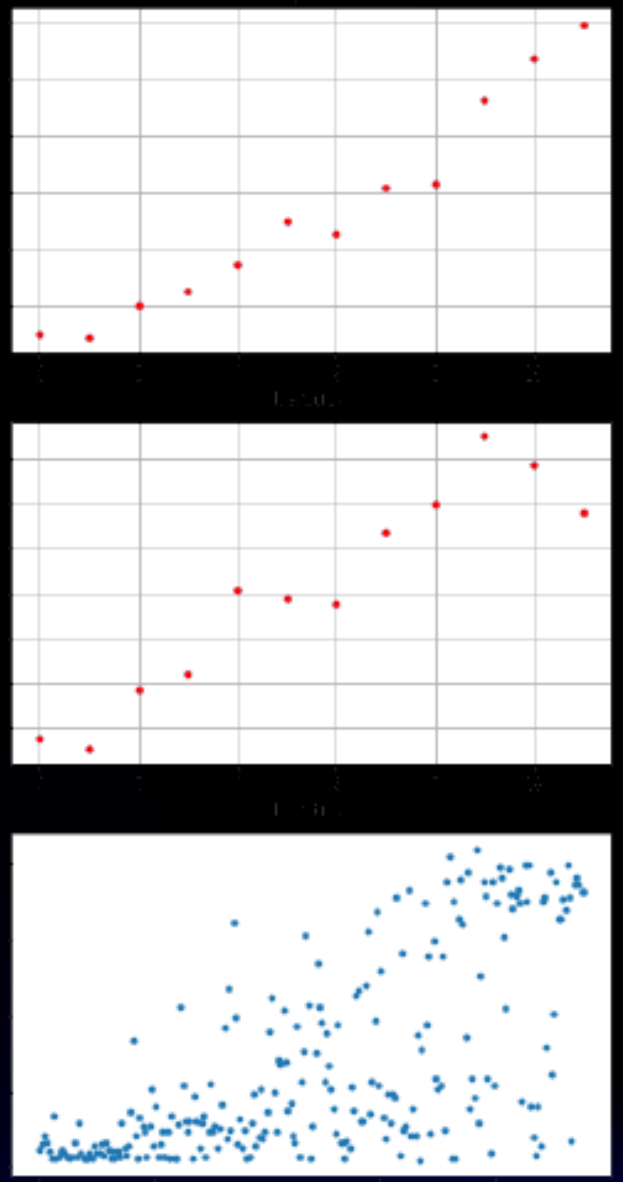
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Rewards per iteration

An iteration is a group of 20 episodes:

For example, iteration 0 is episodes 0–19 and iteration 1 is episodes 20–39

Deviation and scatterplot graphs can help visualize how consistent your model is



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Analyze the reward distribution for your reward function

Note where the reward function is rewarding the model the most

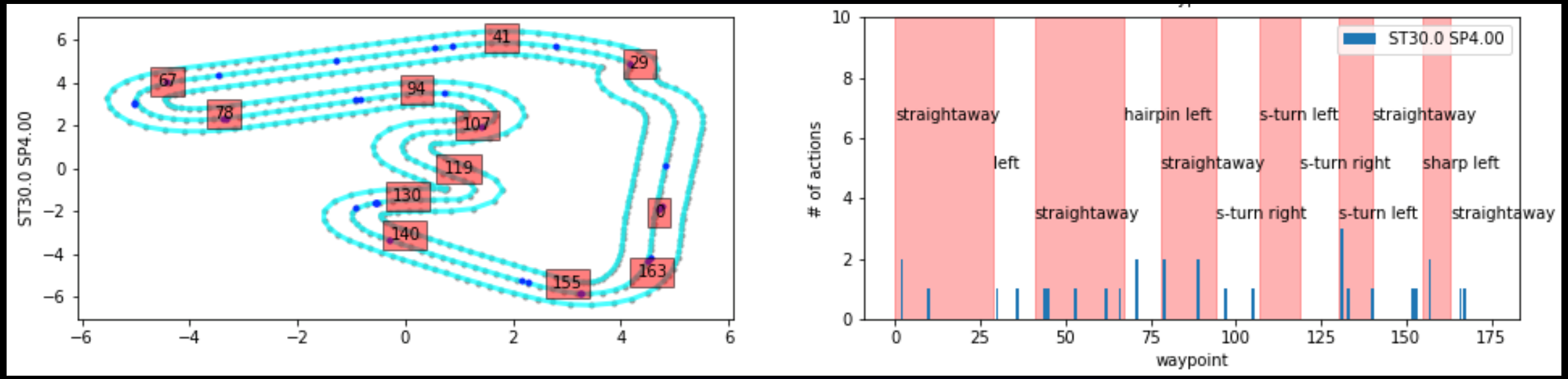
Identify areas not on the optimal racing line as targets for improvement



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Histogram for action distribution for each turn

The notebook has ranges defined for each section of the track; identify which actions are getting rewarded along the track

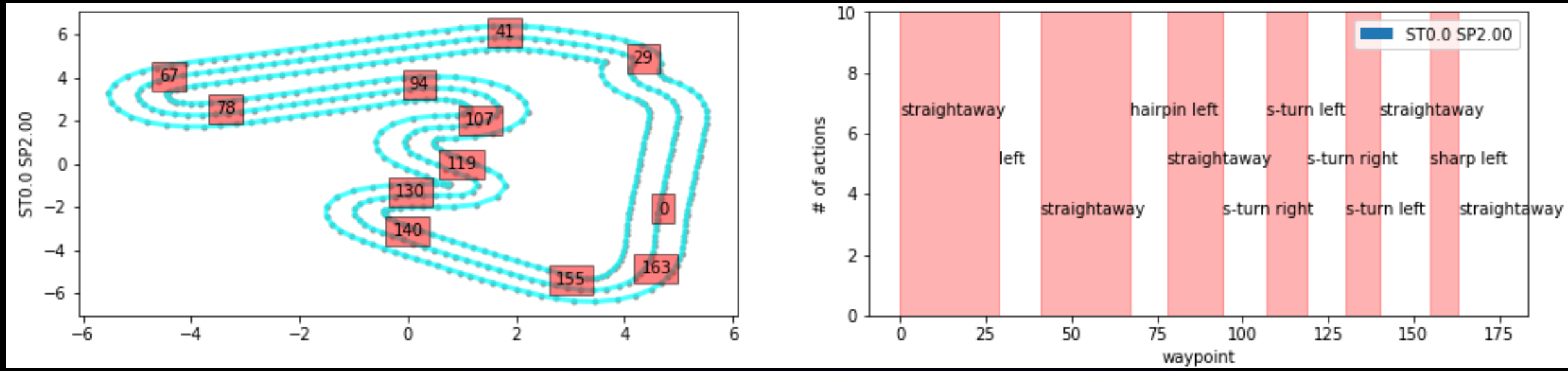


Reward for action right at 30° at 4 m/s

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Histogram for action distribution for each turn

The same training run . . . but a different action. Why is it empty?

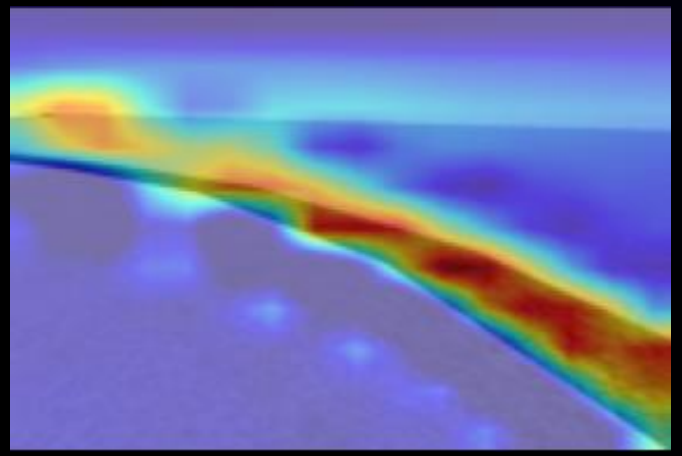


Reward for action straight at 2.0 m/s

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

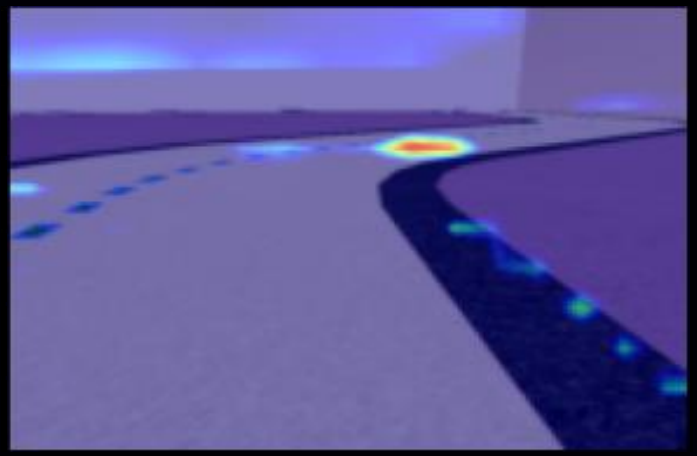
What is the model looking at?

Feed sample images into your model and observe what portions of input are driving the output decision



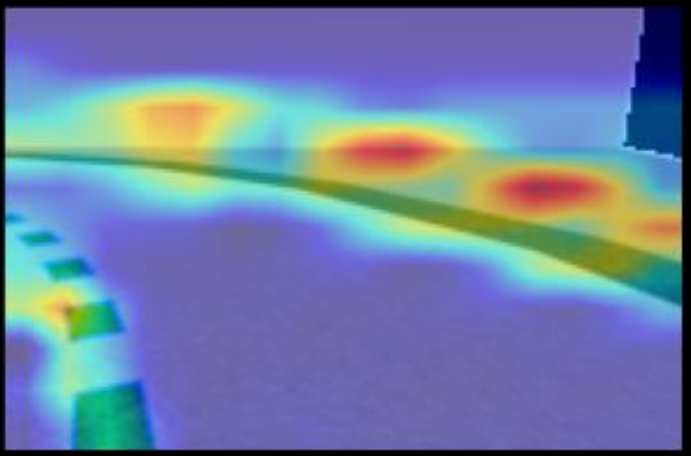
Good

Focusing on outer line



Questionable

Not focusing on anything relevant



Bad

Focusing on the grass and wall

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Recap and wrap-up

- Take advantage of all the reward function parameters available
- Interpret the training graphs to determine if your reward function is delivering the desired results
- Use the training logs and the trained model to get further details into training and model behavior; a good start is the AWS DeepRacer workshops log analysis notebook

- GitHub: https://github.com/aws-samples/aws-deepracer-workshops/
- Tips: https://aws.amazon.com/deepracer/racing-tips/
- Forum: https://forums.aws.amazon.com/forum.jspa?forumID=318

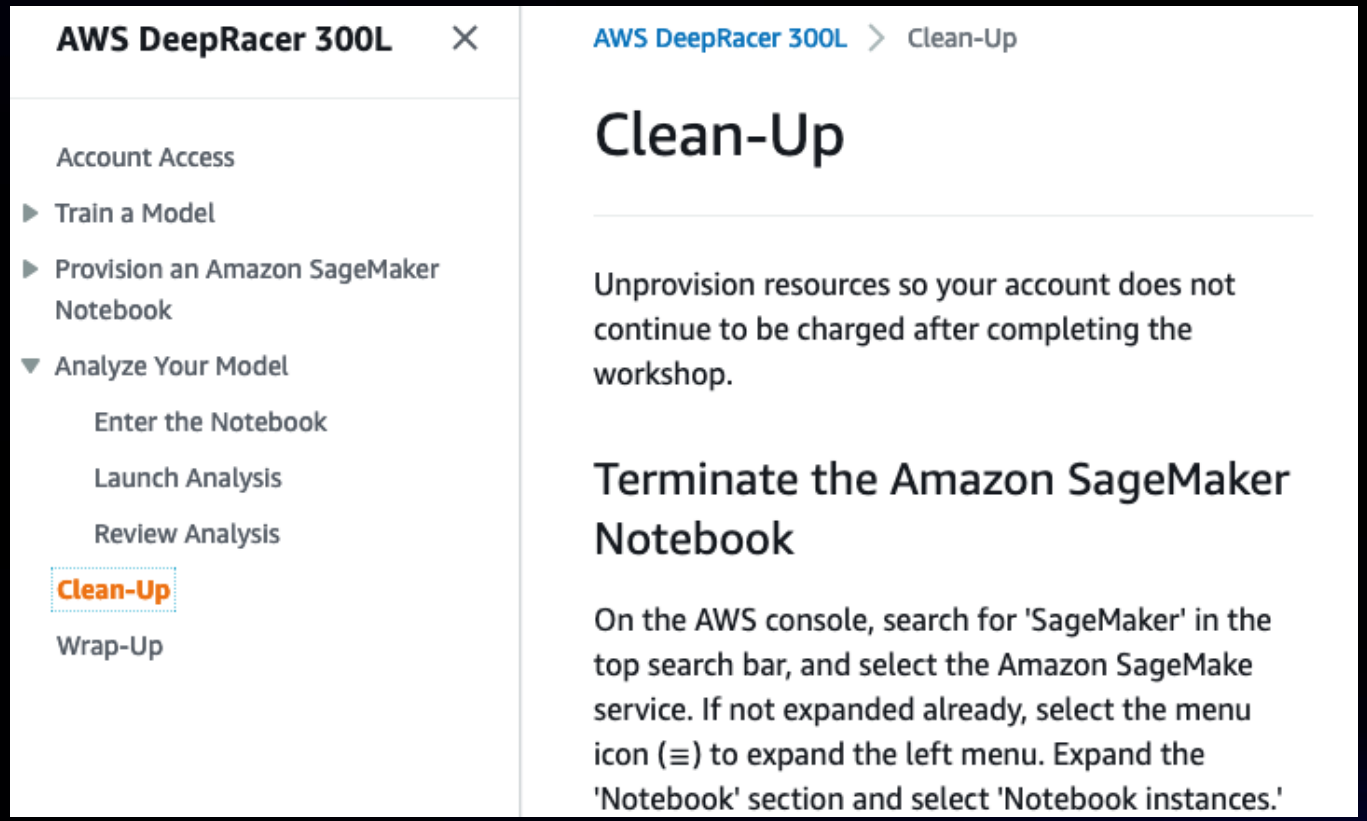
Developers, start your engines!

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Finish with https://deepracer300.workshops.aws.dev/

Don’t forget to clean up!



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Don Barber

donbarb@amazon.com

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.