aws SUMMIT

LONDON | APR 27 2022

ML-06

Speed-up your ML career with AWS DeepRacer

Ananth Balasubramanyam (he/him)
Sr. Solutions Architect
Amazon Web Services

Agenda

AWS DeepRacer origin

RL for the Sunday driver

Virtual simulator

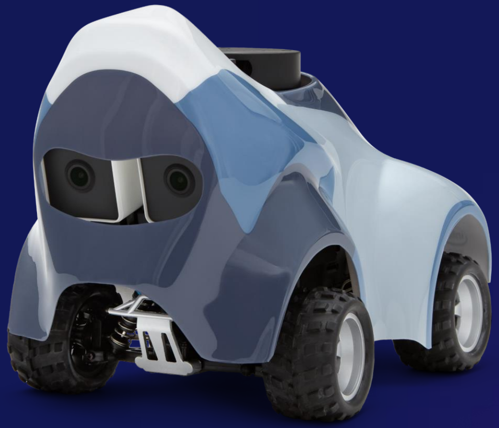
Under the hood

Rubber meets the road

AWS DeepRacer origin

Building your team's skills

AWS DeepLens
Deep Learning

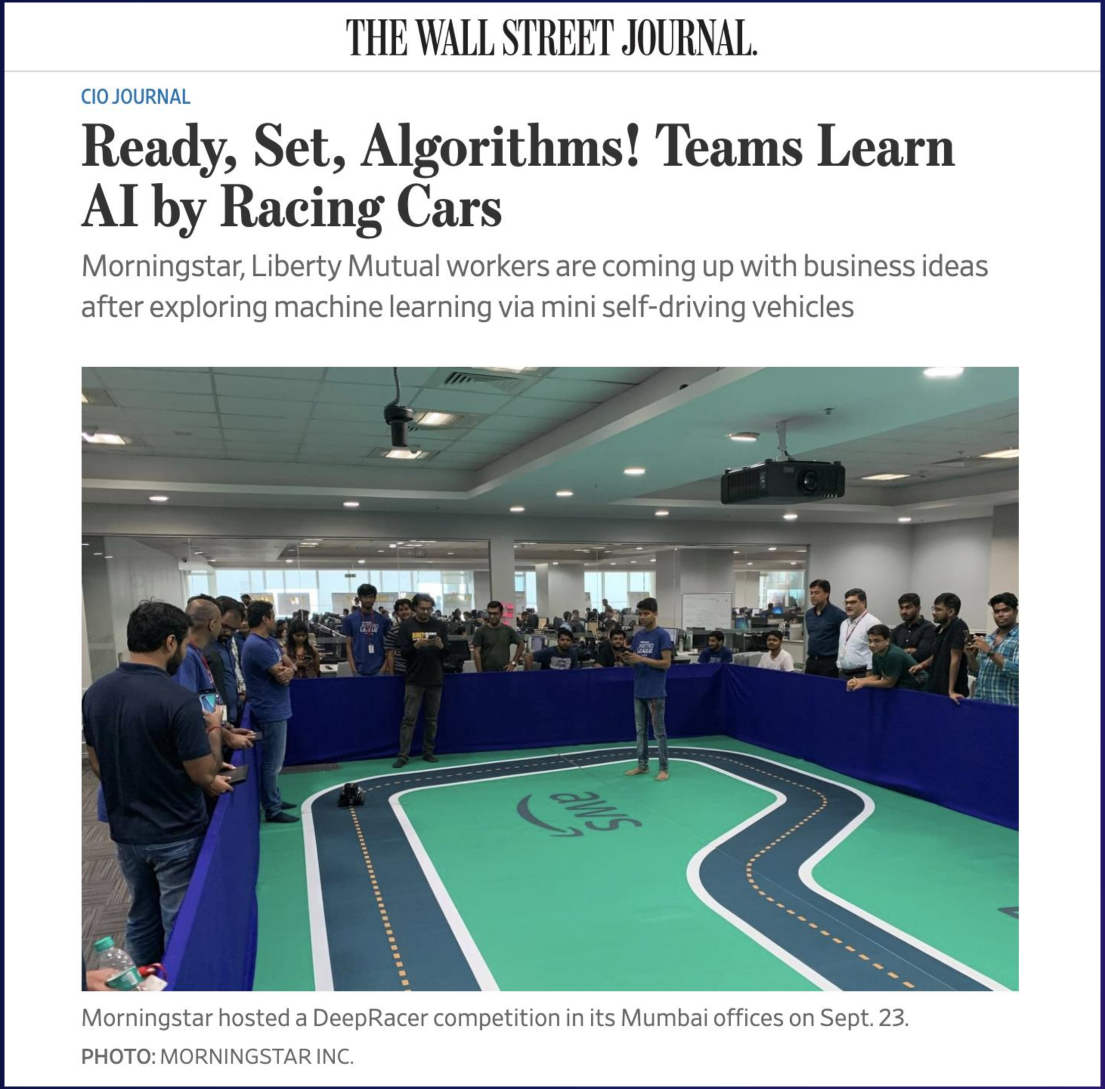

AWS DeepRacer
Reinforcement
Learning

AWS DeepComposer
Generative AI

AWS DeepRacer origin

How can we put reinforcement learning in the hands of all developers? *literally*

AWS DeepRacer is the perfect first date with ML



THE WALL STREET JOURNAL.

CIO JOURNAL

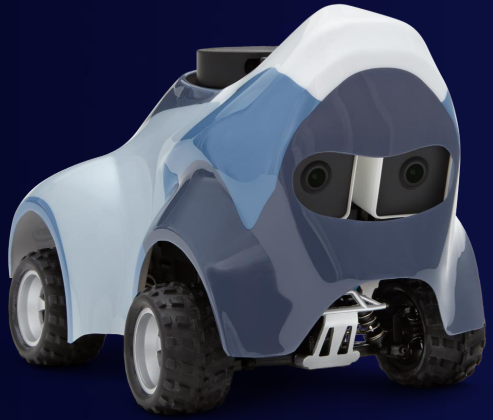
Ready, Set, Algorithms! Teams Learn AI by Racing Cars

Morningstar, Liberty Mutual workers are coming up with business ideas after exploring machine learning via mini self-driving vehicles

Morningstar hosted a DeepRacer competition in its Mumbai offices on Sept. 23.

PHOTO: MORNINGSTAR INC.

AWS DeepRacer: An exciting way for developers to get hands-on experience with reinforcement learning



AWS DeepRacer Evo

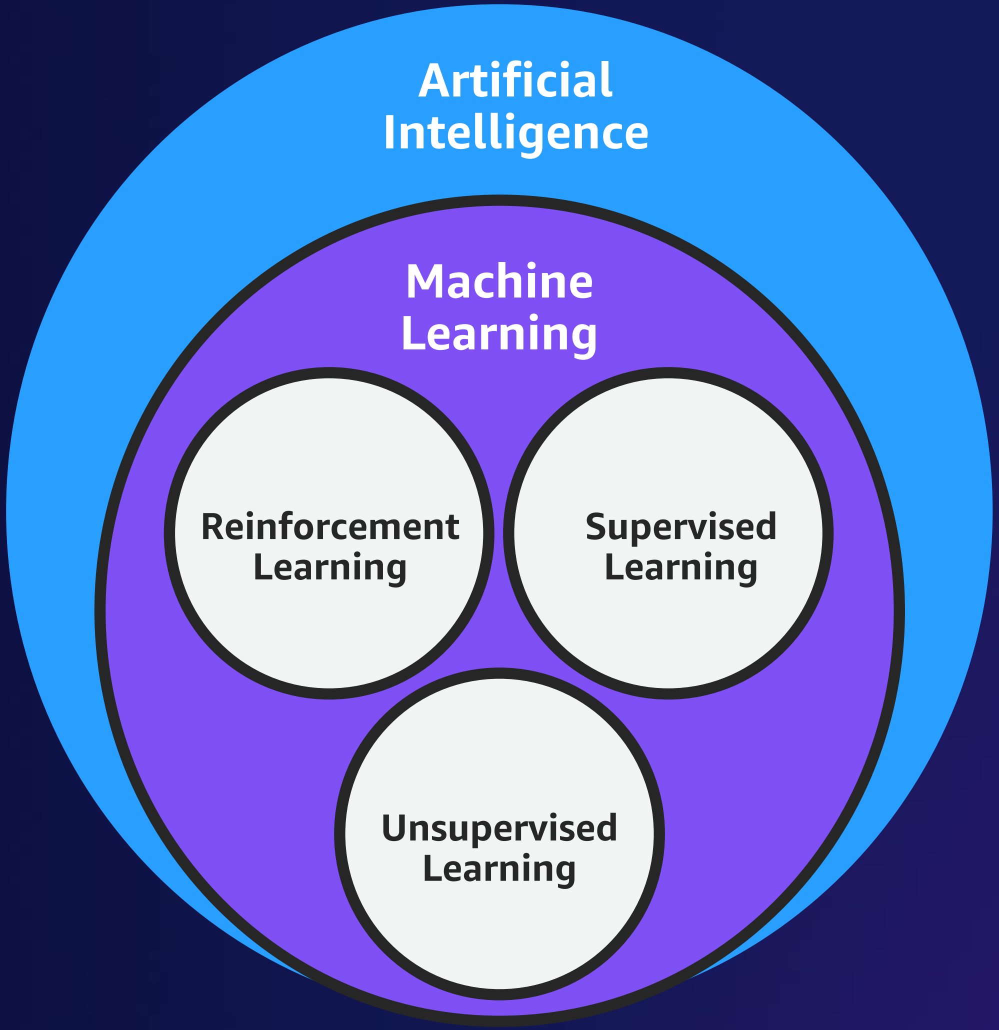
3D racing simulator

Global Racing League

Community Races

RL for the Sunday driver

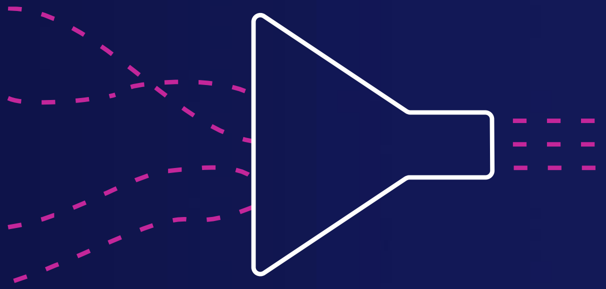
The AWS ML Stack



Machine learning overview

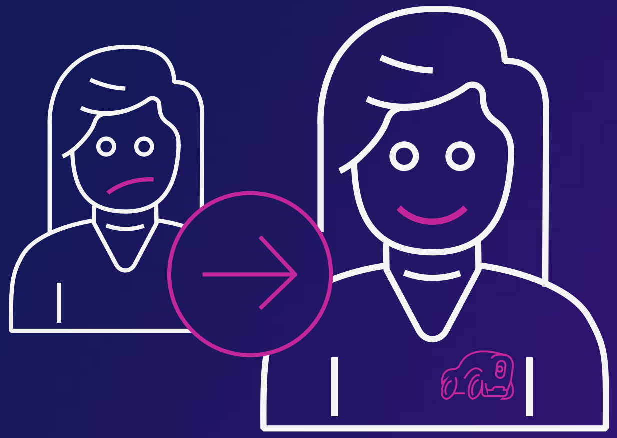
SUPERVISED

Example driven training—every datum has a corresponding label



UNSUPERVISED

No labels for training data



REINFORCEMENT

Learns through consequences of actions in a specific environment

Reinforcement learning in the real world

Reward positive behavior

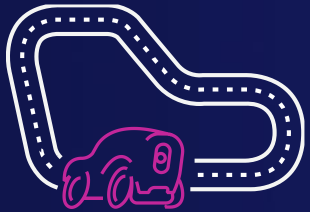
Don’t reward negative behavior

The result!

Reinforcement learning terms

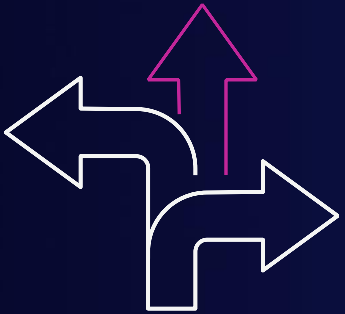
AGENT

ENVIRONMENT



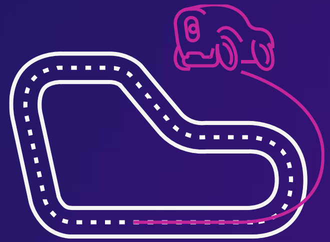
STATE

ACTION



REWARD

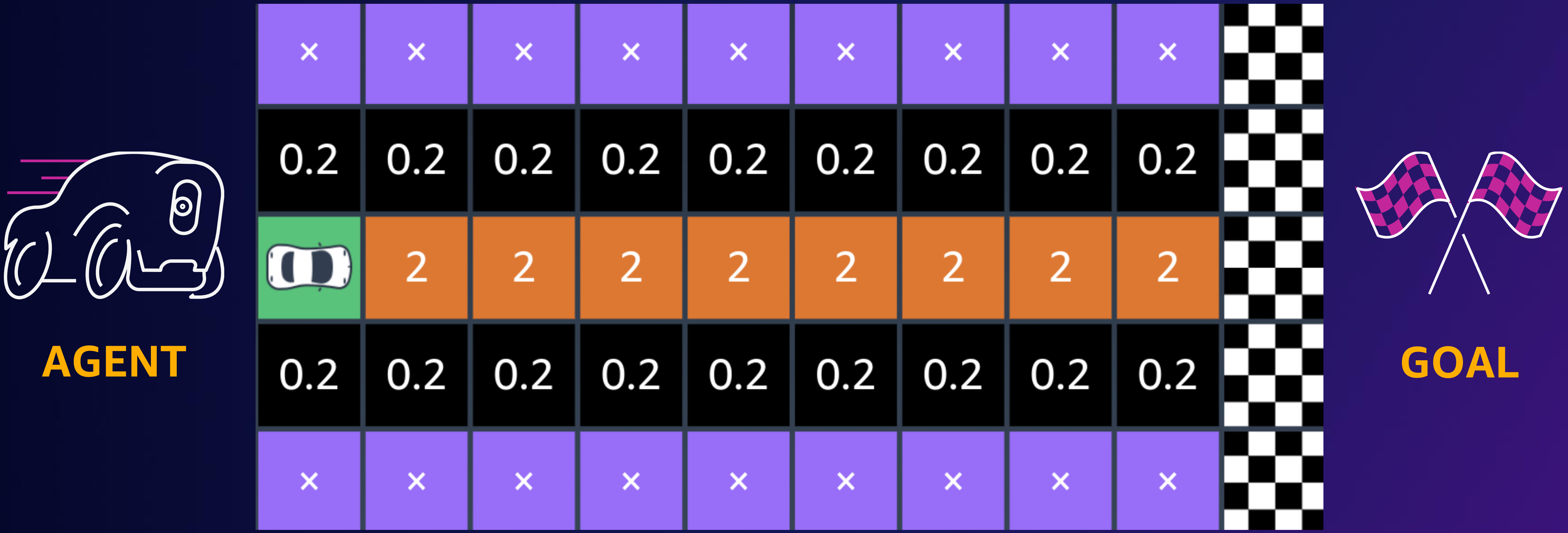
EPISODE



The reward function

The reward function incentivizes particular behaviors and is at the core of reinforcement learning

How to train a reinforcement learning model



×	×	×	×	×	×	×	×	×
0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
	2	2	2	2	2	2	2	2
0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
×	×	×	×	×	×	×	×	×

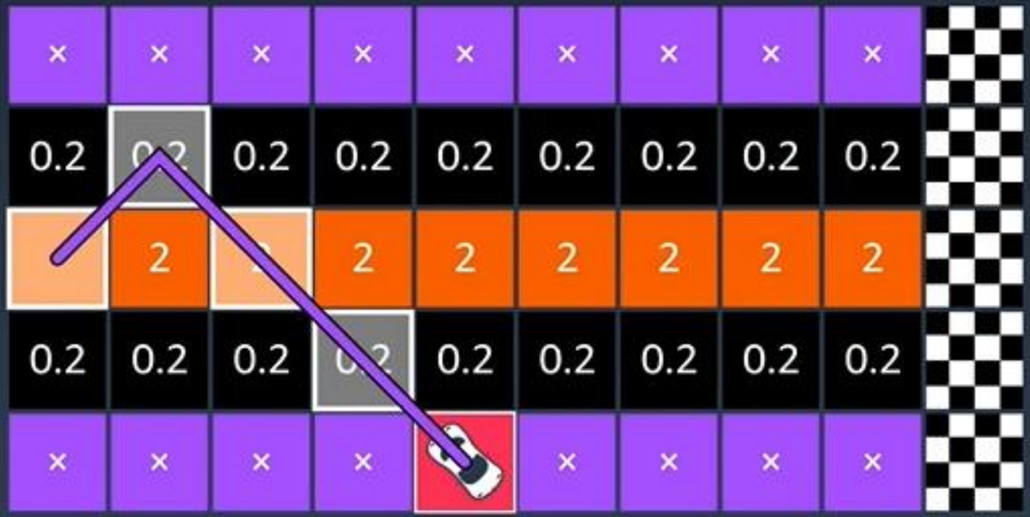
Iteration and convergence

Iteration

Reinforcement learning algorithms are trained by repeated optimization of cumulative rewards.

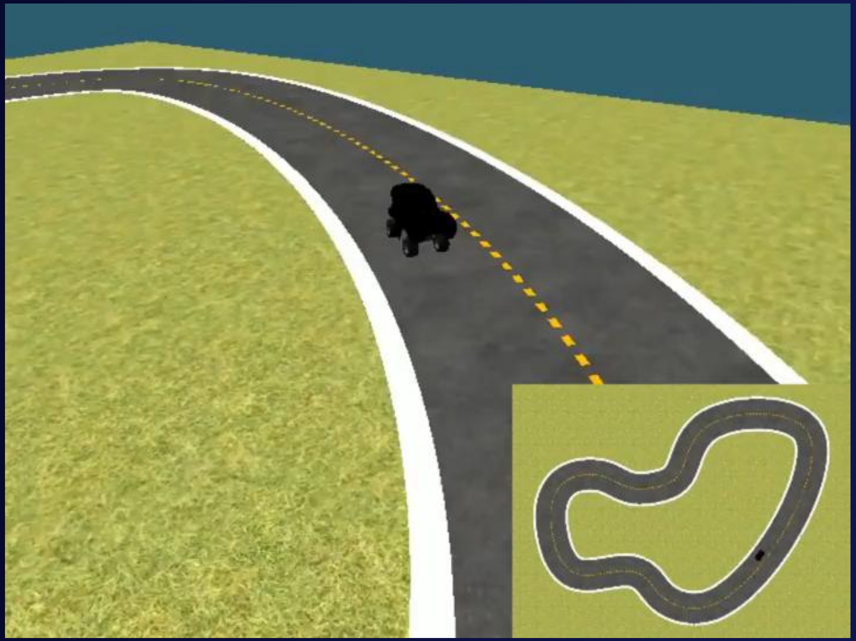
The model will learn which action (and then subsequent actions) will result in the highest cumulative reward on the way to the goal.

Learning doesn't just happen on the first go; it takes some iteration. First, the agent needs to explore and see where it can get the highest rewards, before it can exploit that knowledge.

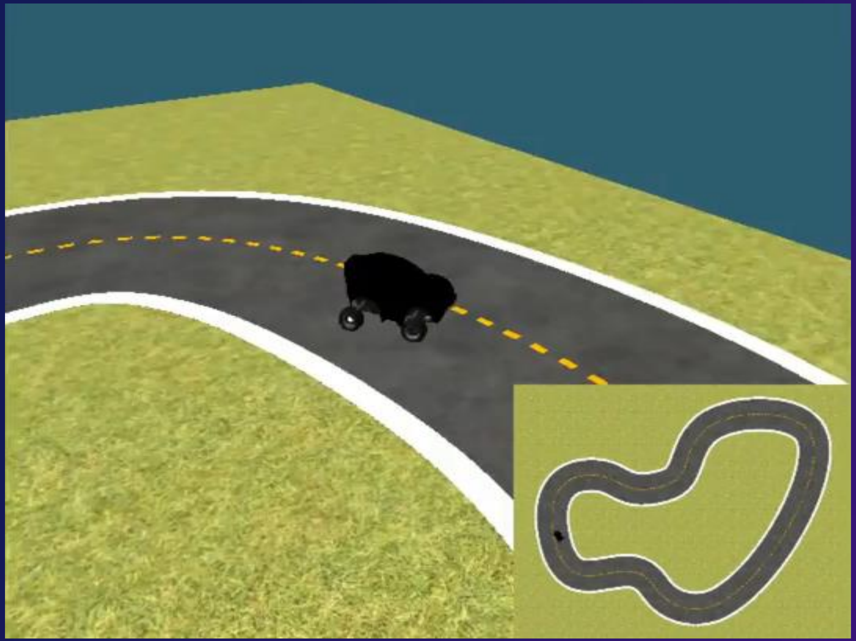


Exploration vs. Exploitation

Exploration

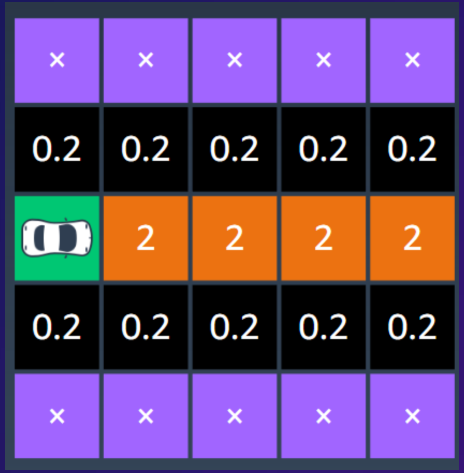


Exploitation



Recap before we continue

Reinforcement learning



×	×	×	×	×
0.2	0.2	0.2	0.2	0.2
🚗	2	2	2	2
0.2	0.2	0.2	0.2	0.2
×	×	×	×	×

AWS DeepRacer problem formulation

MODEL

AGENT

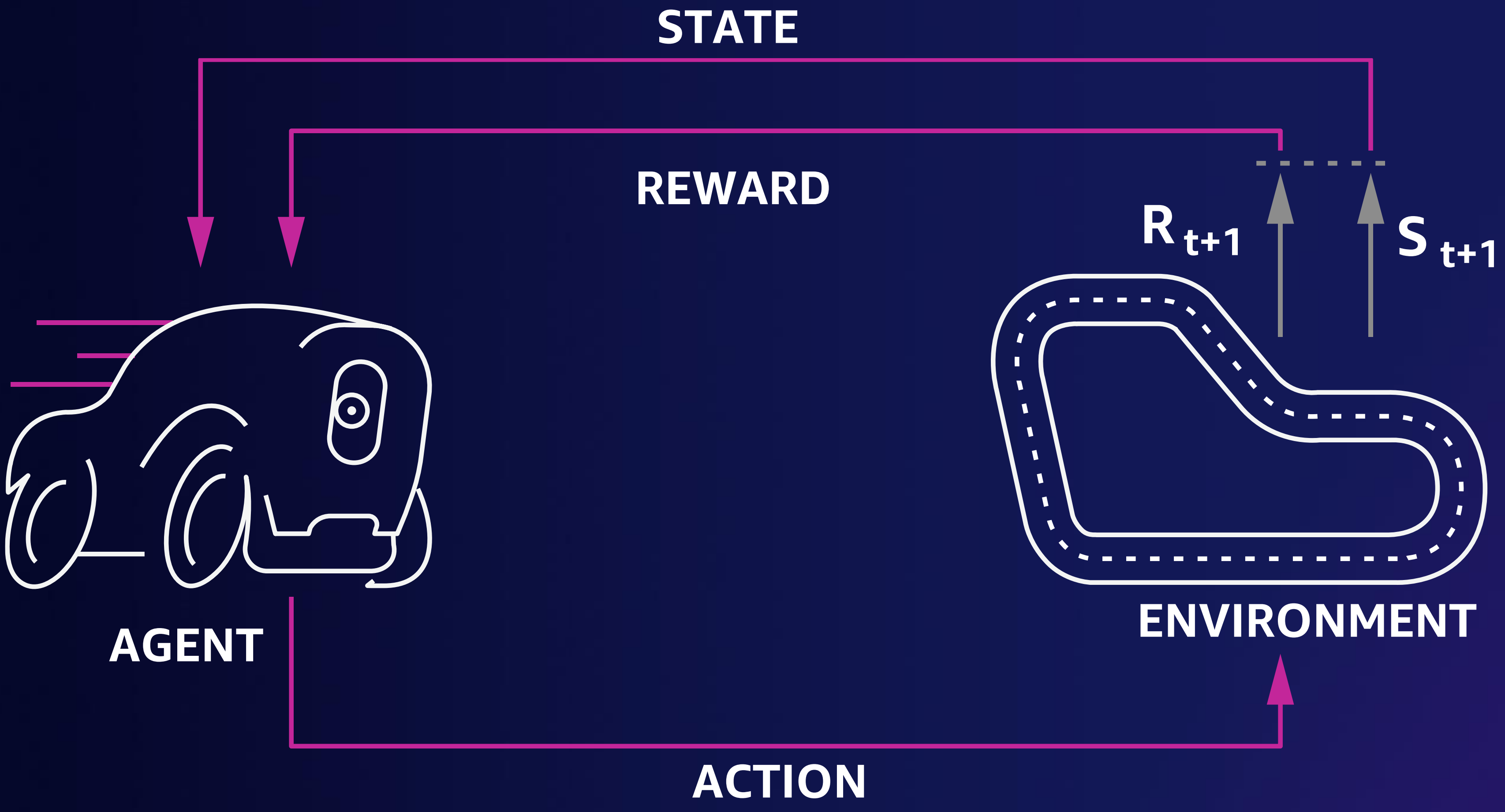
ACTION

STATE

ENVIRONMENT

GOAL

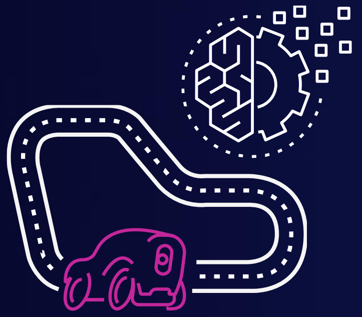
How does learning happen?



VALUE FUNCTION

POLICY FUNCTION

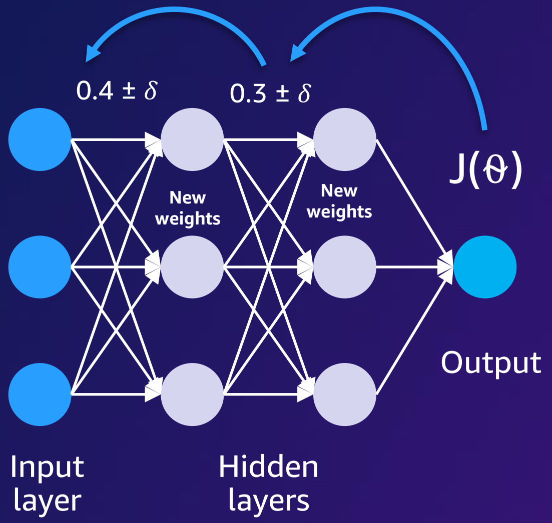
RL algorithms: Vanilla policy gradient



Gather episode using current policy

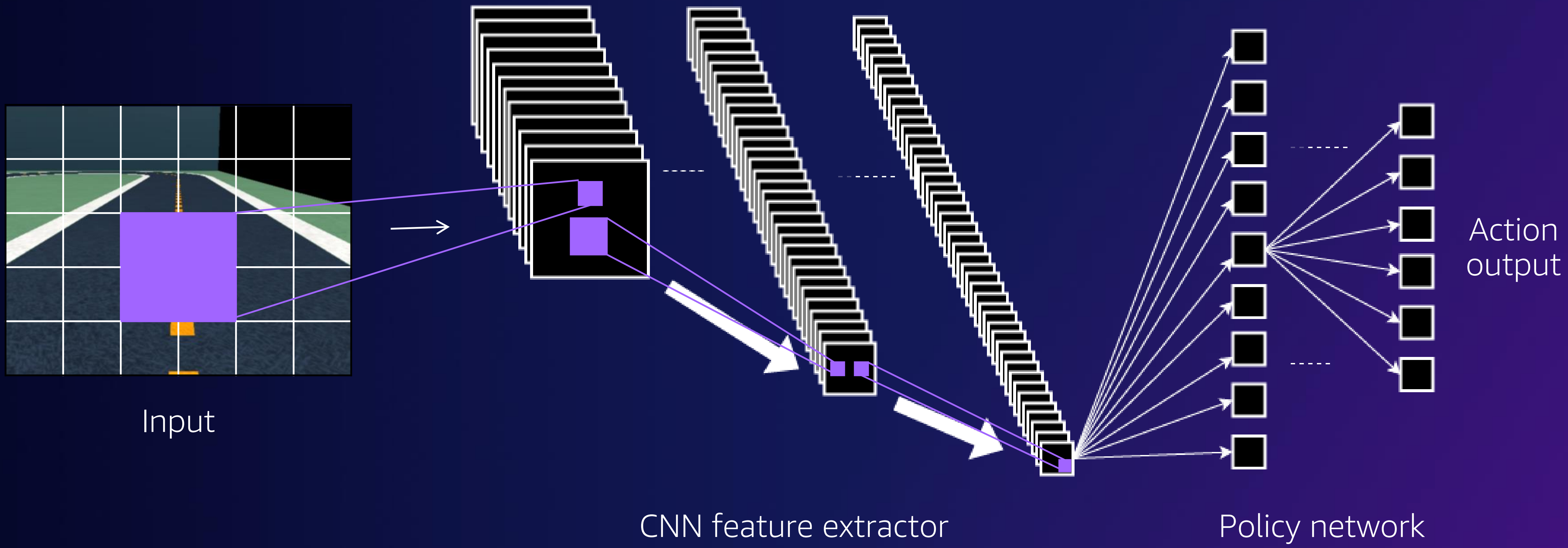
Goal: Maximum cumulative rewards

Calculate the gradient of est. cumulative reward



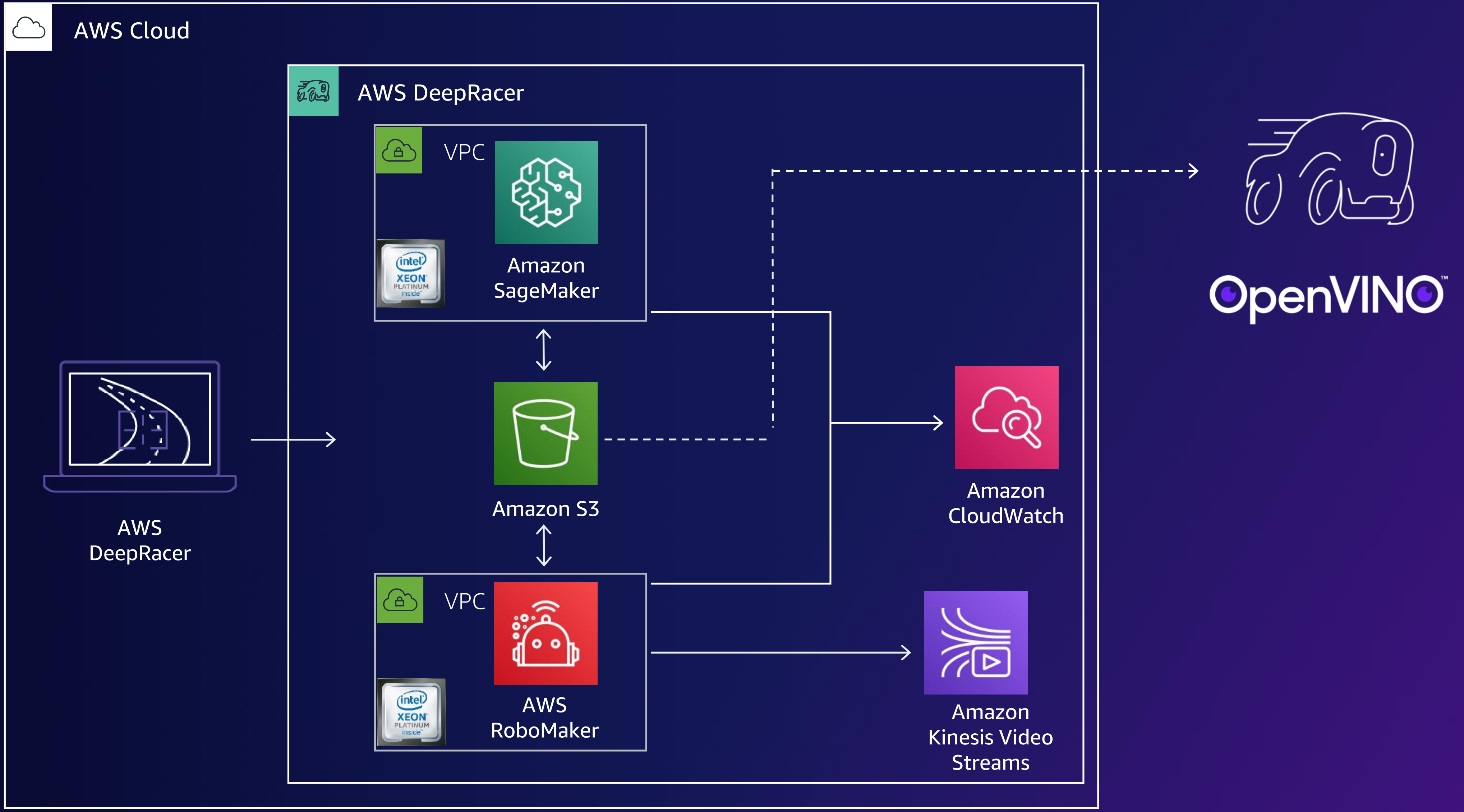
Update the policy for gradient ascent

AWS DeepRacer neural network architecture

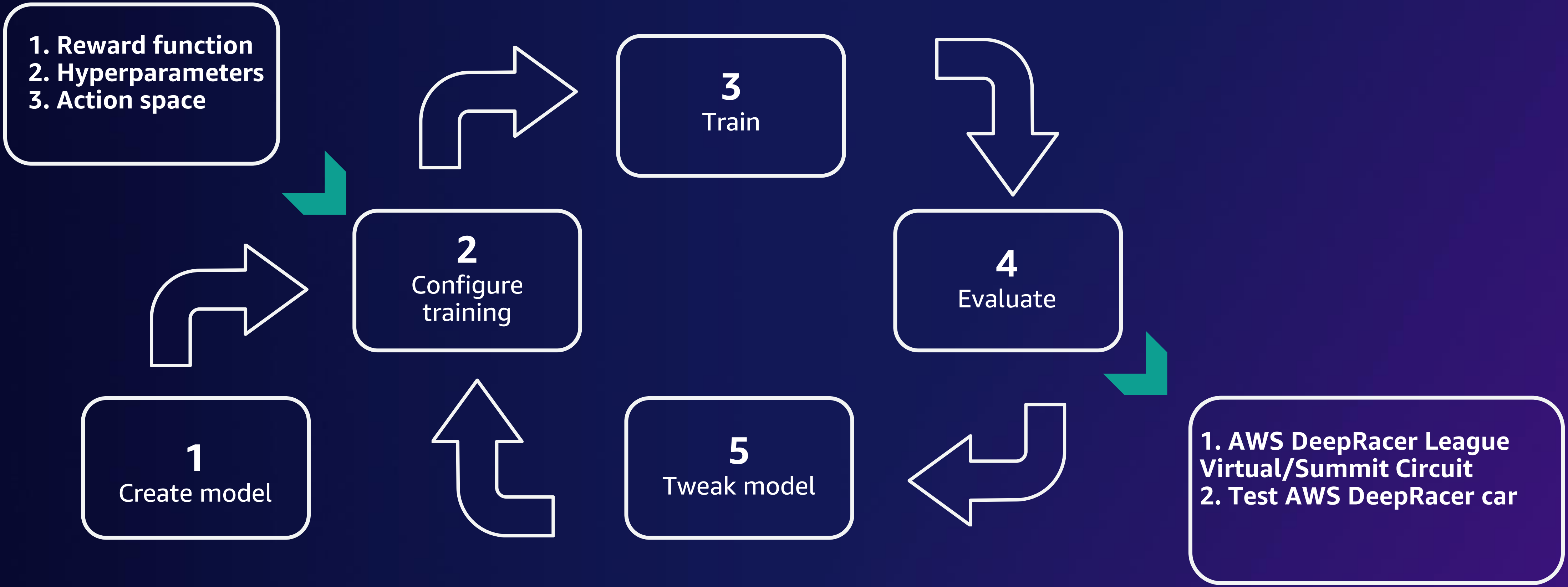


Virtual simulator

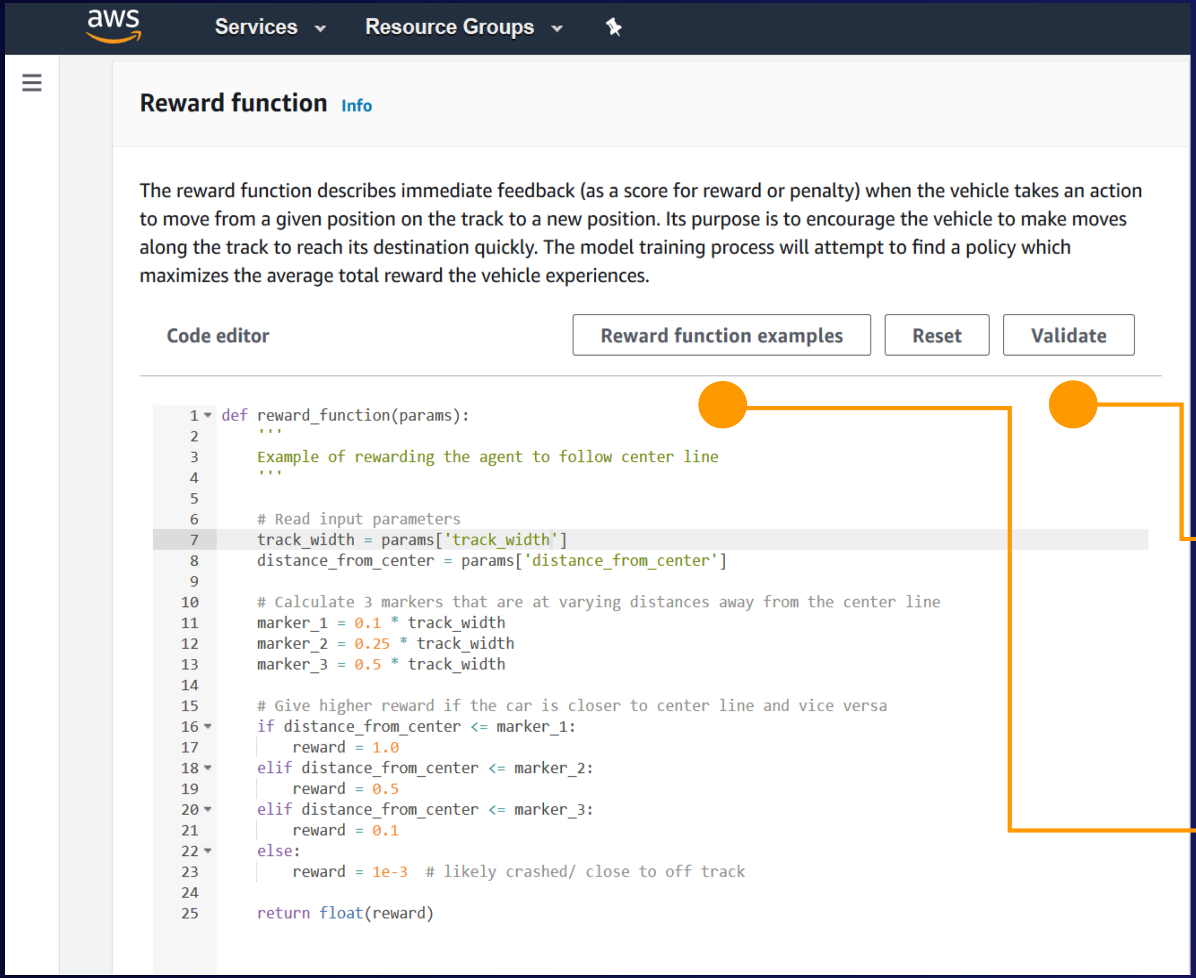
AWS DeepRacer simulator architecture



AWS DeepRacer console diagram



Programming your own reward function

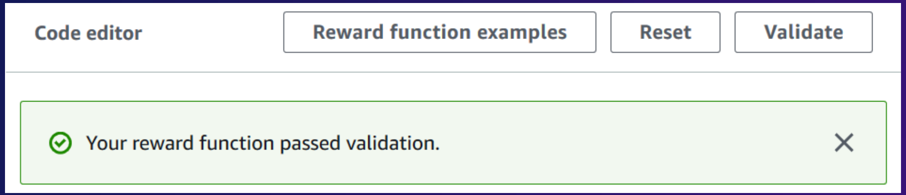


The reward function describes immediate feedback (as a score for reward or penalty) when the vehicle takes an action to move from a given position on the track to a new position. Its purpose is to encourage the vehicle to make moves along the track to reach its destination quickly. The model training process will attempt to find a policy which maximizes the average total reward the vehicle experiences.

```python
def reward_function(params):
    '''
    Example of rewarding the agent to follow center line
    '''

    # Read input parameters
    track_width = params['track_width']
    distance_from_center = params['distance_from_center']

    # Calculate 3 markers that are at varying distances away from the center line
    marker_1 = 0.1 * track_width
    marker_2 = 0.25 * track_width
    marker_3 = 0.5 * track_width

    # Give higher reward if the car is closer to center line and vice versa
    if distance_from_center <= marker_1:
        reward = 1.0
    elif distance_from_center <= marker_2:
        reward = 0.5
    elif distance_from_center <= marker_3:
        reward = 0.1
    else:
        reward = 1e-3  # likely crashed/ close to off track

    return float(reward)
```

Code editor: Python 3 Syntax

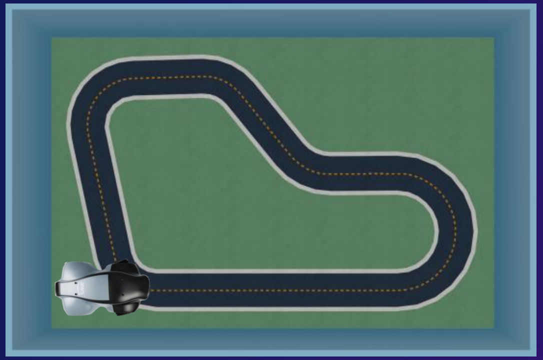
Code validation via AWS Lambda

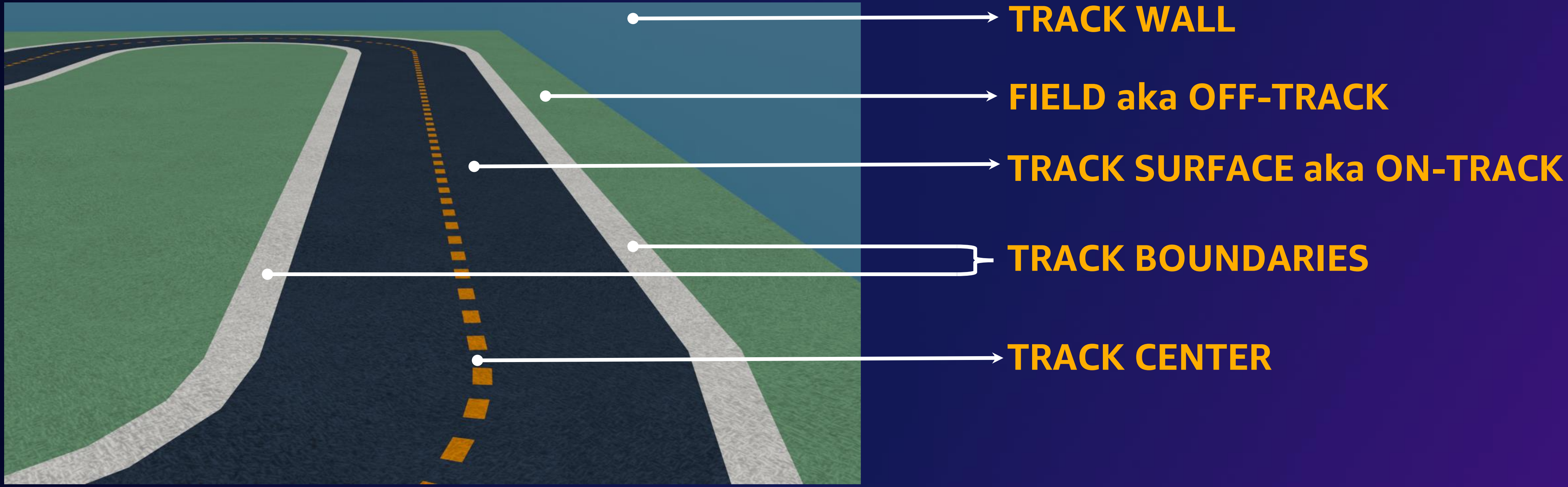


Your reward function passed validation.

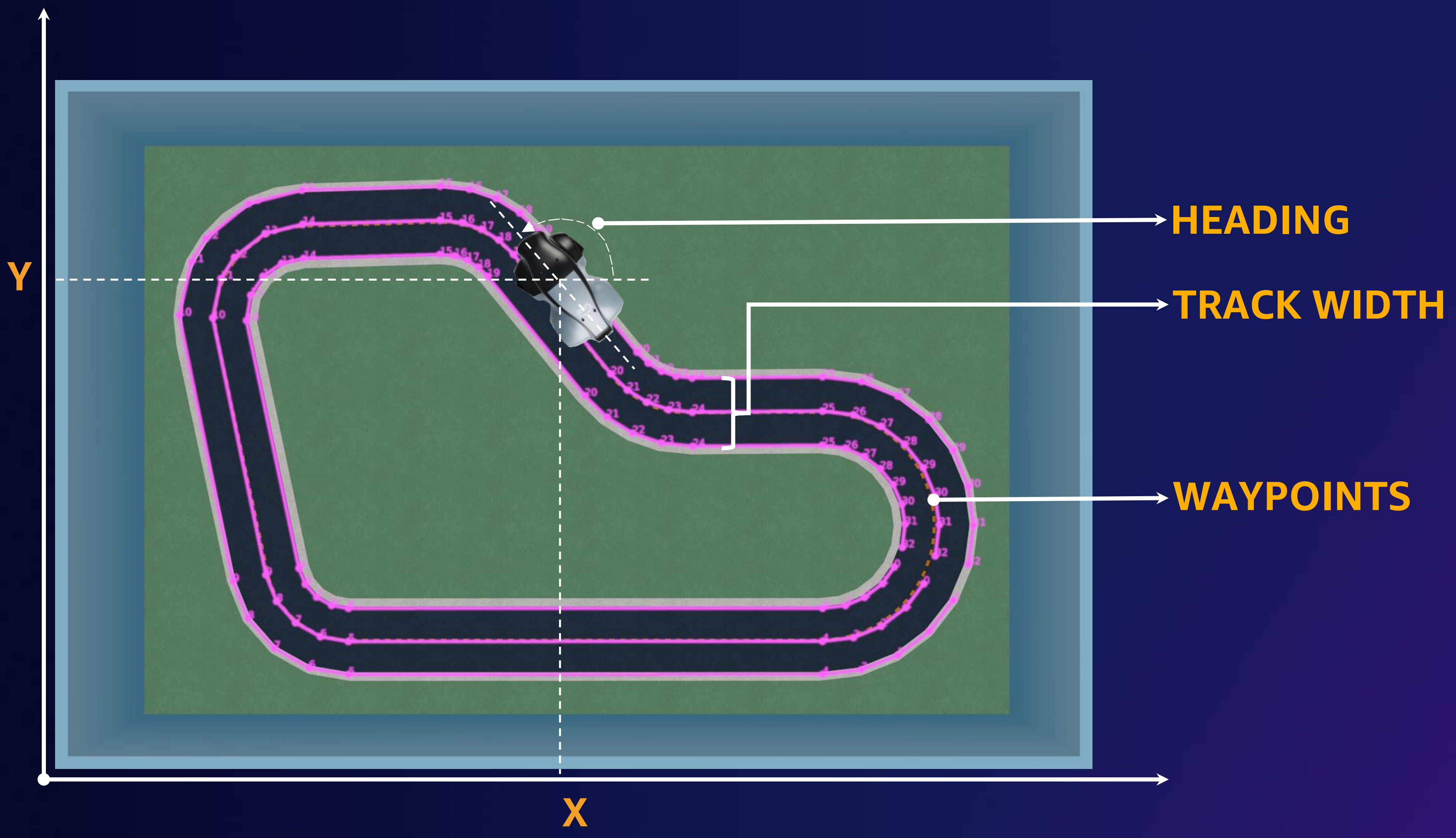
Three example reward functions

Track components

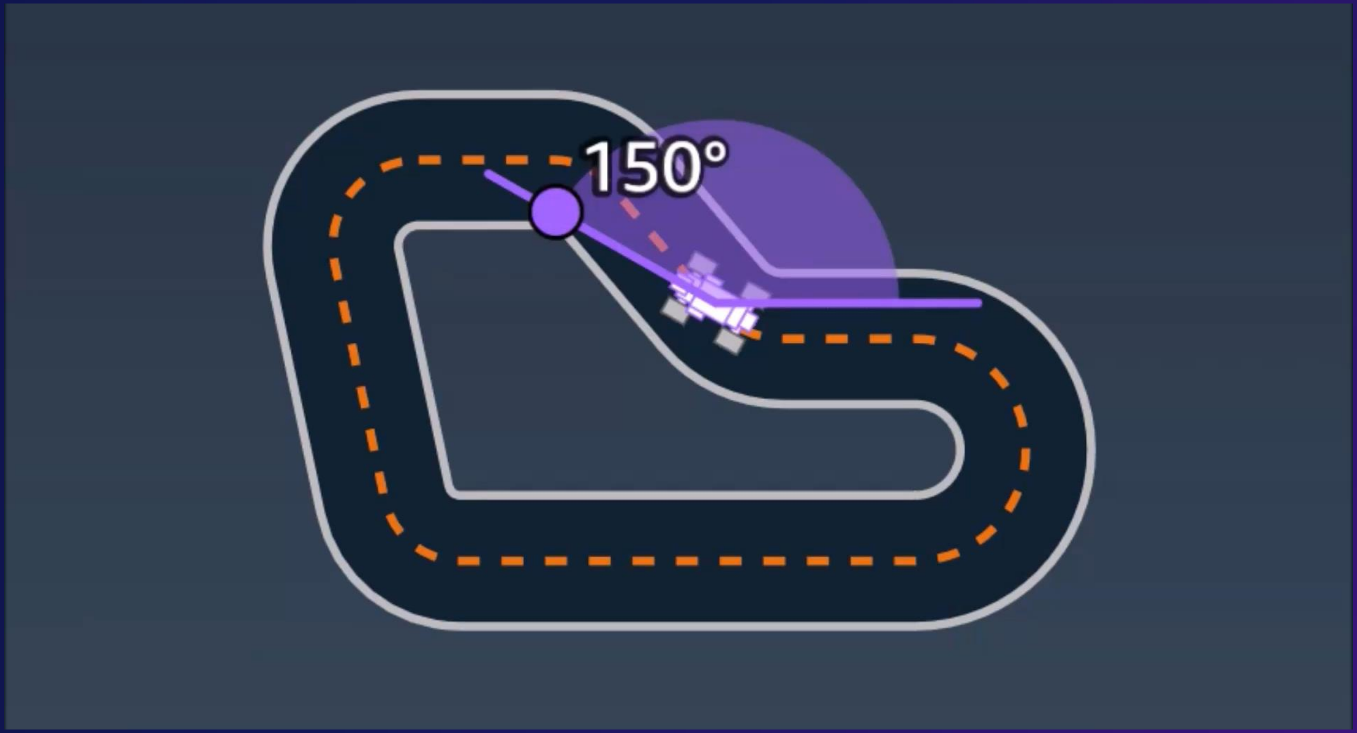




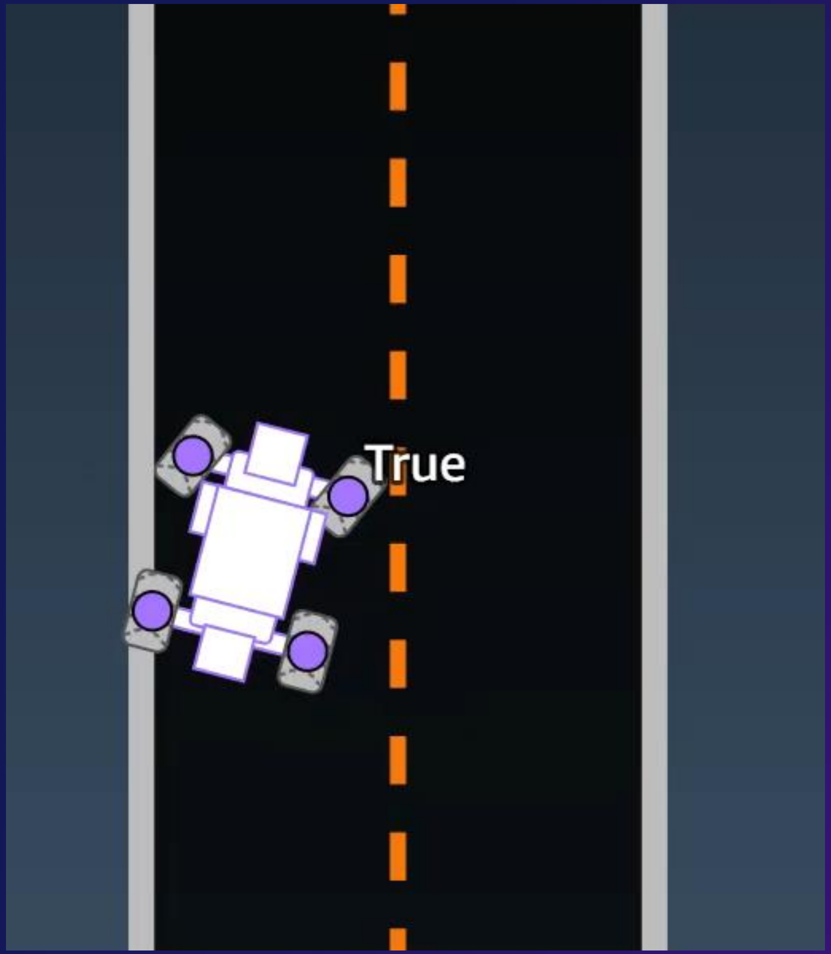
Coordinate system and track waypoints



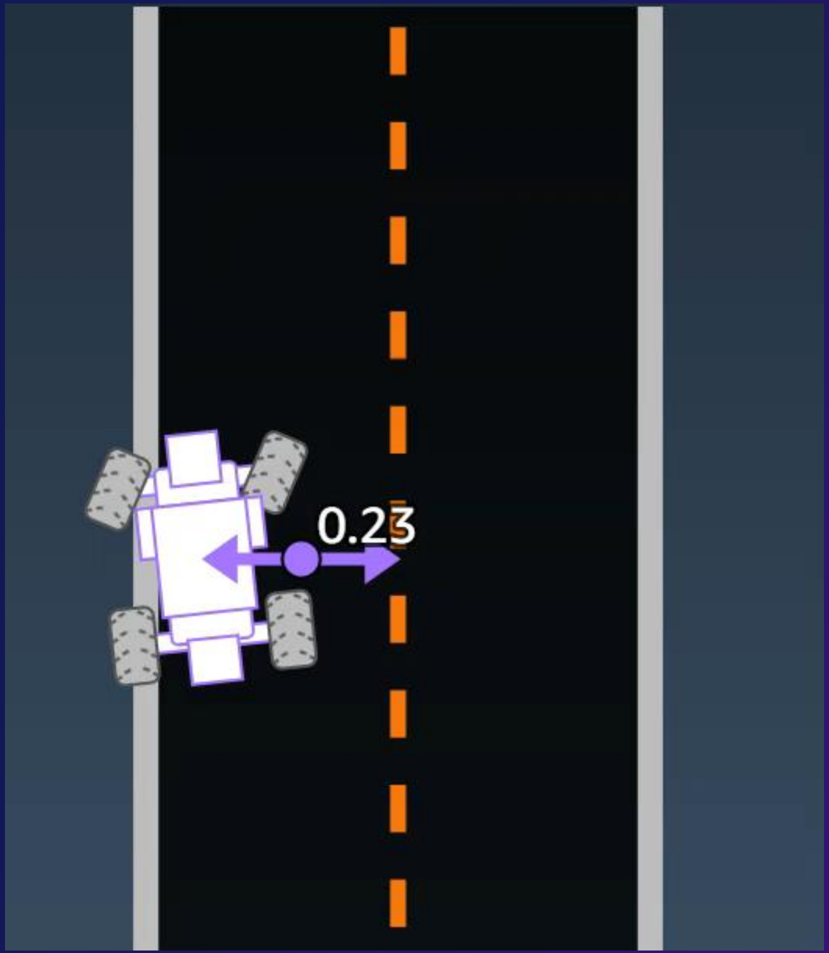
Example parameter: heading



Example parameter: all_wheels_on_track



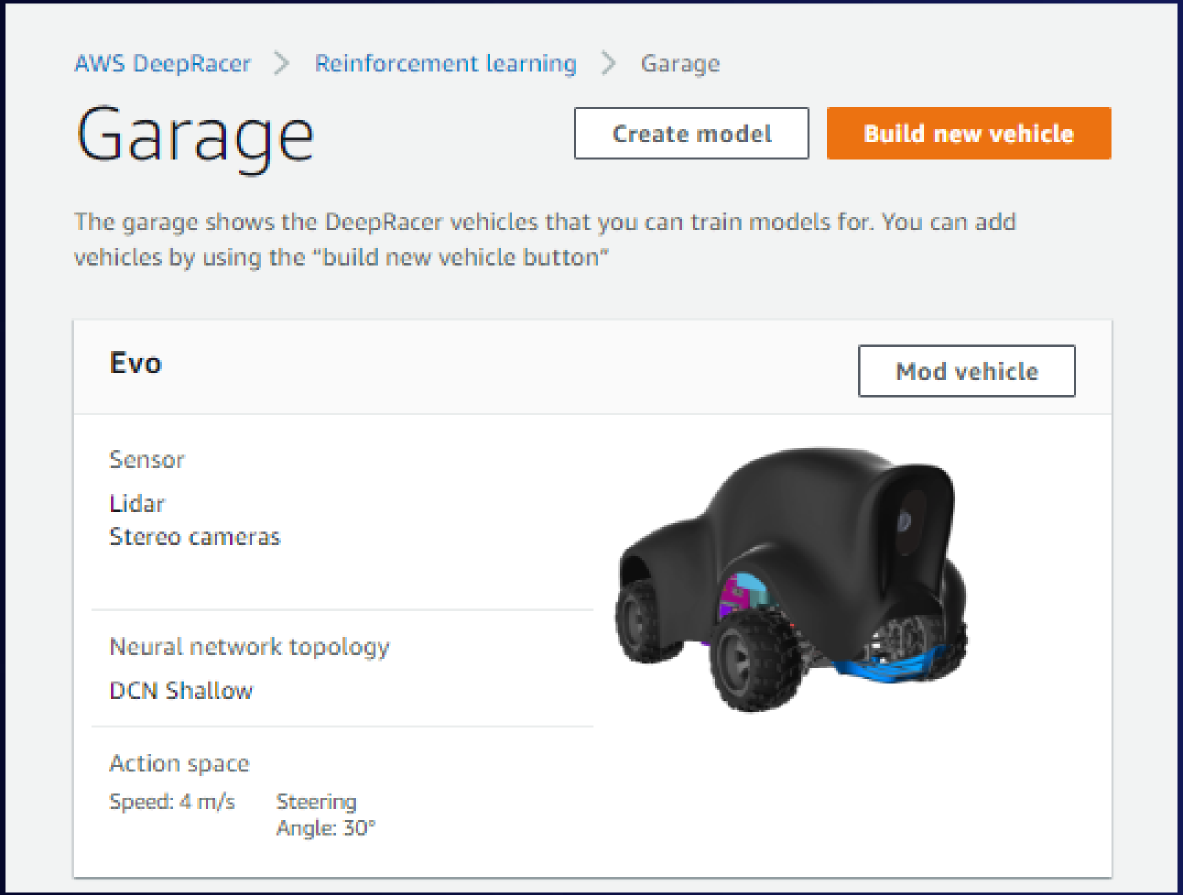
Example parameter: distance_from_center



AWS DeepRacer in the console

https://console.aws.amazon.com/deepracer/

Customize your agent's sensors in the Garage



AWS DeepRacer > Reinforcement learning > Garage

Garage

The garage shows the DeepRacer vehicles that you can train models for. You can add vehicles by using the "build new vehicle button"

Evo

Sensor

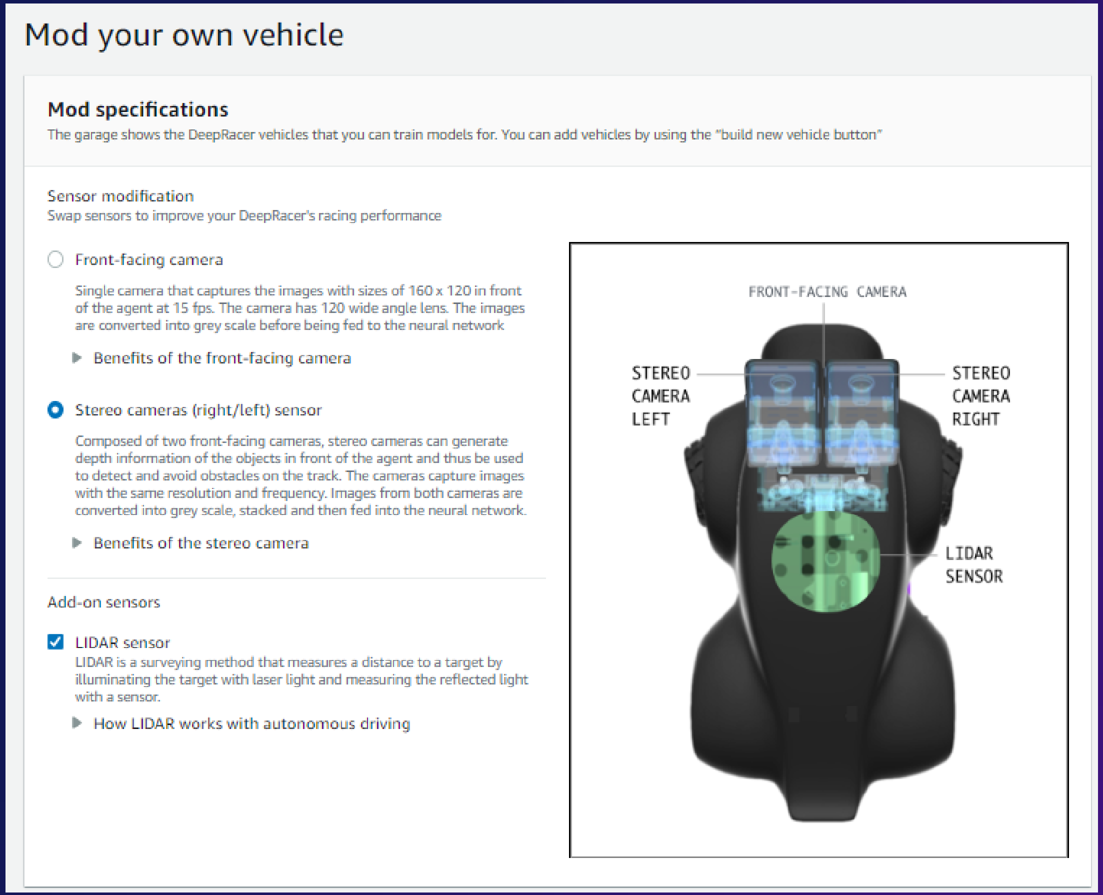
Lidar
Stereo cameras

Neural network topology

DCN Shallow

Action space

Speed: 4 m/s   Steering Angle: 30°



Mod your own vehicle

Mod specifications

The garage shows the DeepRacer vehicles that you can train models for. You can add vehicles by using the "build new vehicle button"

Sensor modification

Swap sensors to improve your DeepRacer's racing performance

○ Front-facing camera

Single camera that captures the images with sizes of 160 x 120 in front of the agent at 15 fps. The camera has 120 wide angle lens. The images are converted into grey scale before being fed to the neural network

▶ Benefits of the front-facing camera

● Stereo cameras (right/left) sensor

Composed of two front-facing cameras, stereo cameras can generate depth information of the objects in front of the agent and thus be used to detect and avoid obstacles on the track. The cameras capture images with the same resolution and frequency. Images from both cameras are converted into grey scale, stacked and then fed into the neural network.

▶ Benefits of the stereo camera

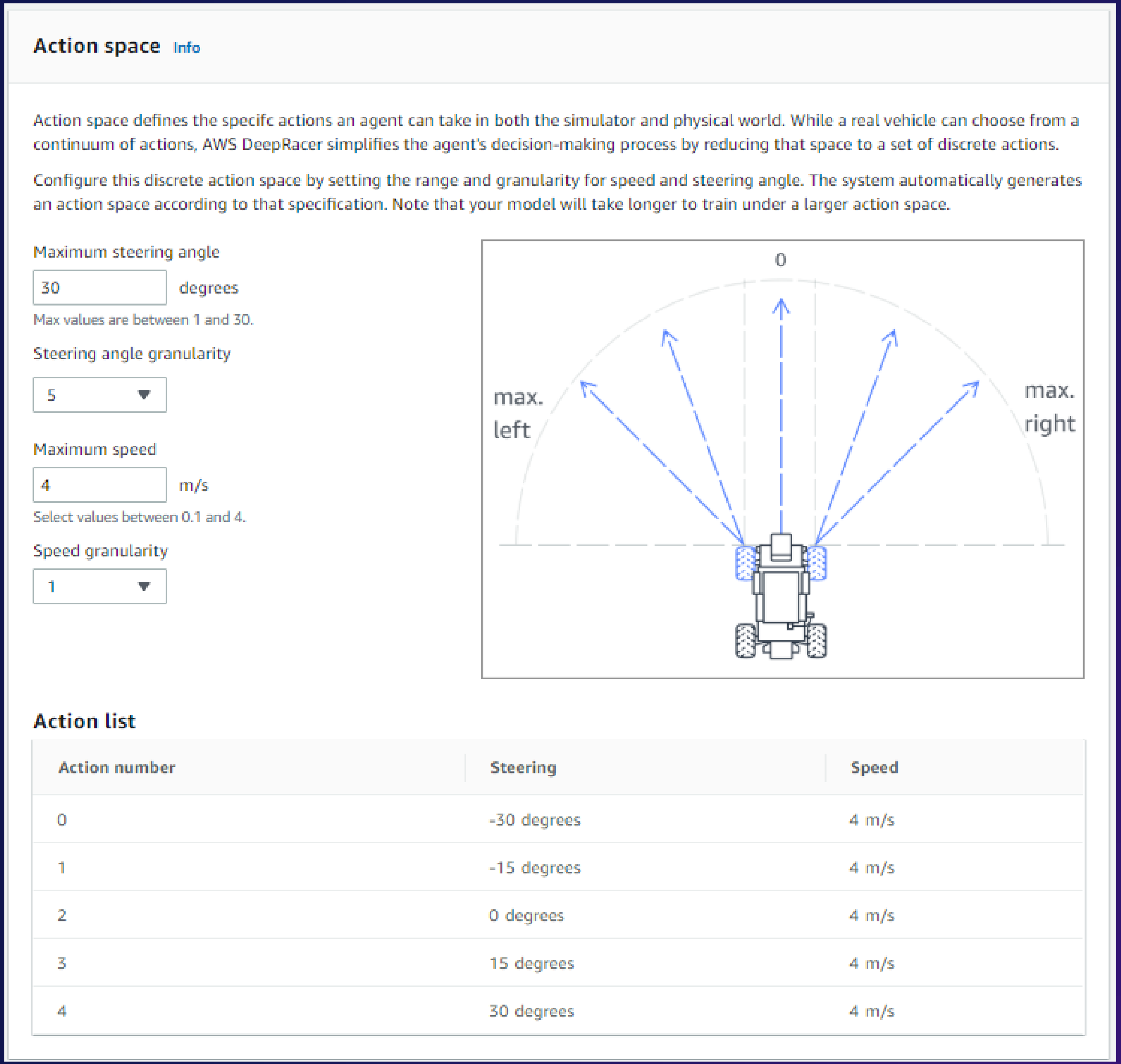
Add-on sensors

✓ LIDAR sensor

LIDAR is a surveying method that measures a distance to a target by illuminating the target with laser light and measuring the reflected light with a sensor.

▶ How LIDAR works with autonomous driving

Action space



Action space Info

Action space defines the specifc actions an agent can take in both the simulator and physical world. While a real vehicle can choose from a continuum of actions, AWS DeepRacer simplifies the agent's decision-making process by reducing that space to a set of discrete actions.

Configure this discrete action space by setting the range and granularity for speed and steering angle. The system automatically generates an action space according to that specification. Note that your model will take longer to train under a larger action space.

Maximum steering angle: 30 degrees
Max values are between 1 and 30.

Steering angle granularity: 5

Maximum speed: 4 m/s
Select values between 0.1 and 4.

Speed granularity: 1

Action list

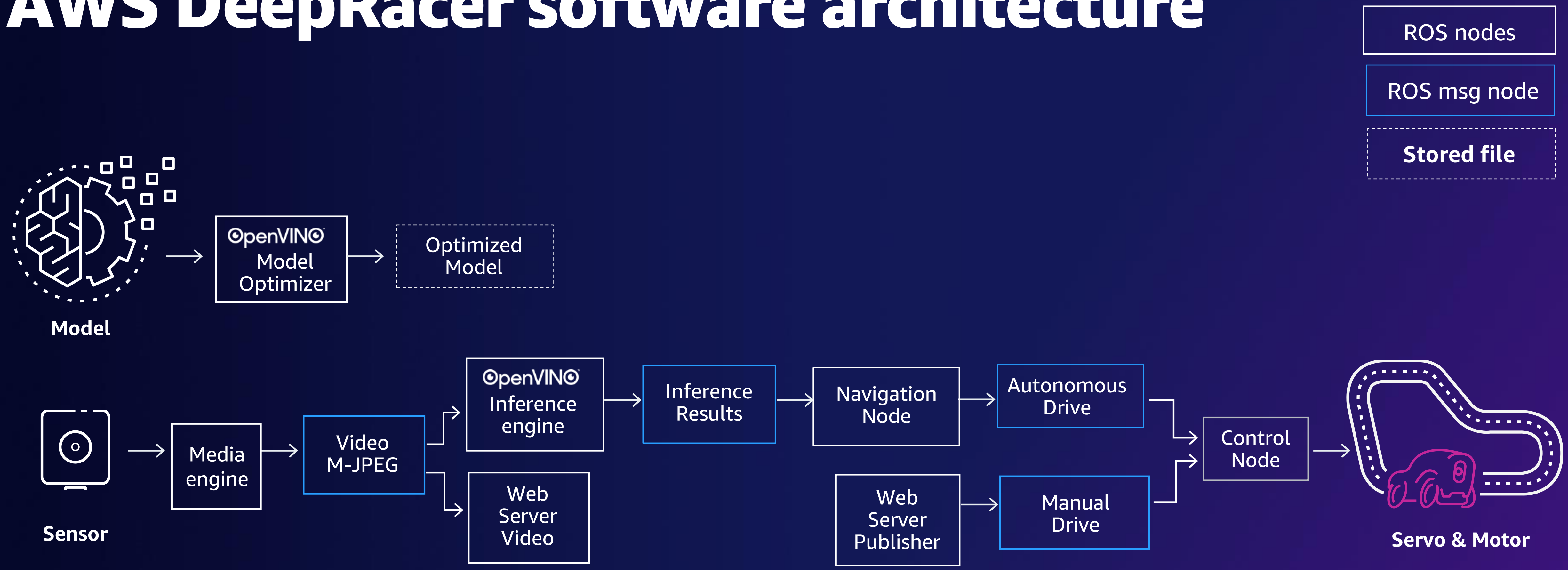
Action number	Steering	Speed
0	-30 degrees	4 m/s
1	-15 degrees	4 m/s
2	0 degrees	4 m/s
3	15 degrees	4 m/s
4	30 degrees	4 m/s

AWS DeepRacer – Under the hood

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

AWS DeepRacer software architecture



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

Rubber meets the road

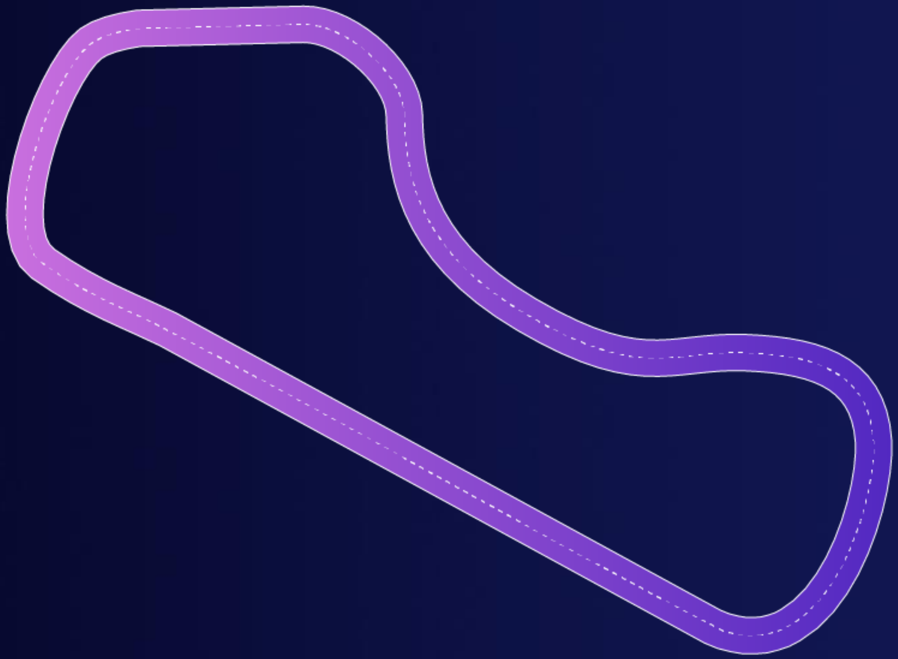
Race for prizes and glory in the AWS DeepRacer League

- Train your AWS DeepRacer model and compete
  - Online in the Virtual Circuit
  - In person in the Summit Circuit (visit the Expo Hall)

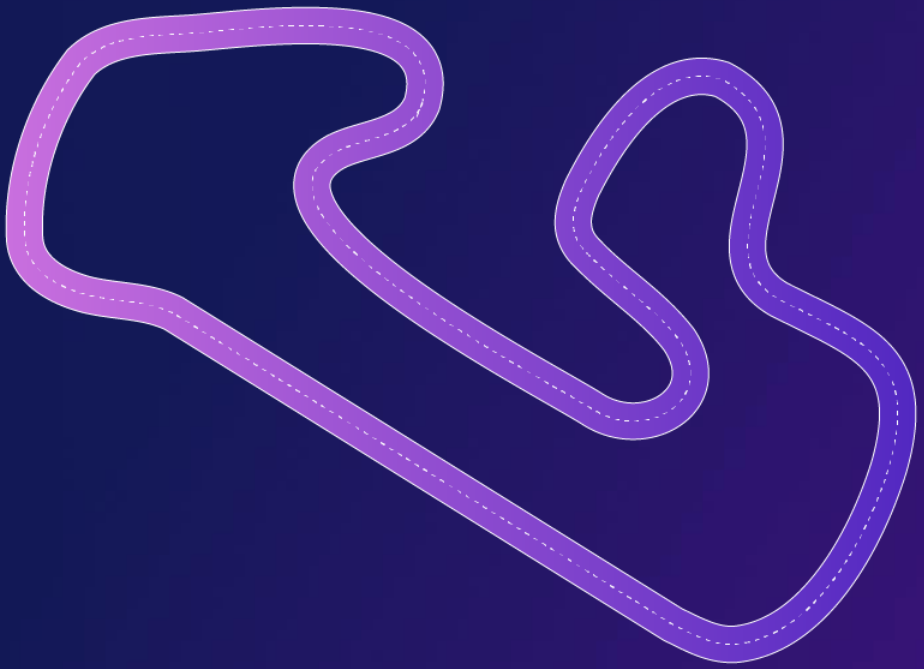
www.deepracerleague.com

Join and compete in the 2022 League

Open Division



Pro Division



Additional resources

- DeepRacer Slack Community: http://join.deepracing.io/
- GitHub: https://github.com/aws-samples/aws-deepracer-workshops/
- Free video course: https://www.aws.training/Details/eLearning?id=32143
- Tips: https://aws.amazon.com/deepracer/racing-tips/
- Forum: https://forums.aws.amazon.com/forum.jspa?forumID=318
- Intel® Distribution of OpenVINO™ toolkit: https://software.intel.com/en-us/openvino-toolkit

Learn in-demand AWS Cloud skills

AWS Skill Builder

Access **500+ free** digital courses and Learning Plans

Explore resources with a variety of skill levels and **16+** languages to meet your learning needs

Deepen your skills with digital learning on demand

Train now

AWS Certifications

Earn an industry-recognized credential

Receive Foundational, Associate, Professional, and Specialty certifications

Join the **AWS Certified community** and get exclusive benefits

Access **new** exam guides

Thank you!

Ananth Balasubramanyam

ananthrb (linkedin)

Please complete the session survey