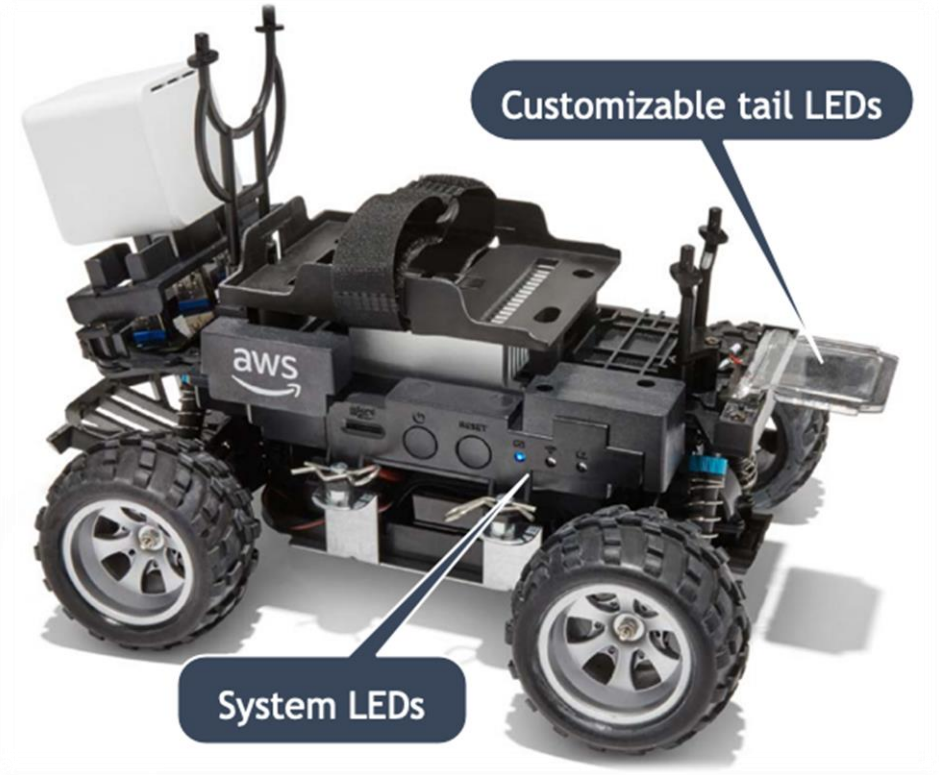
AWS DEEPRACER

What is an AWS DeepRacer?

Source: https://www.youtube.com/watch?v=vCt-F2HscOU

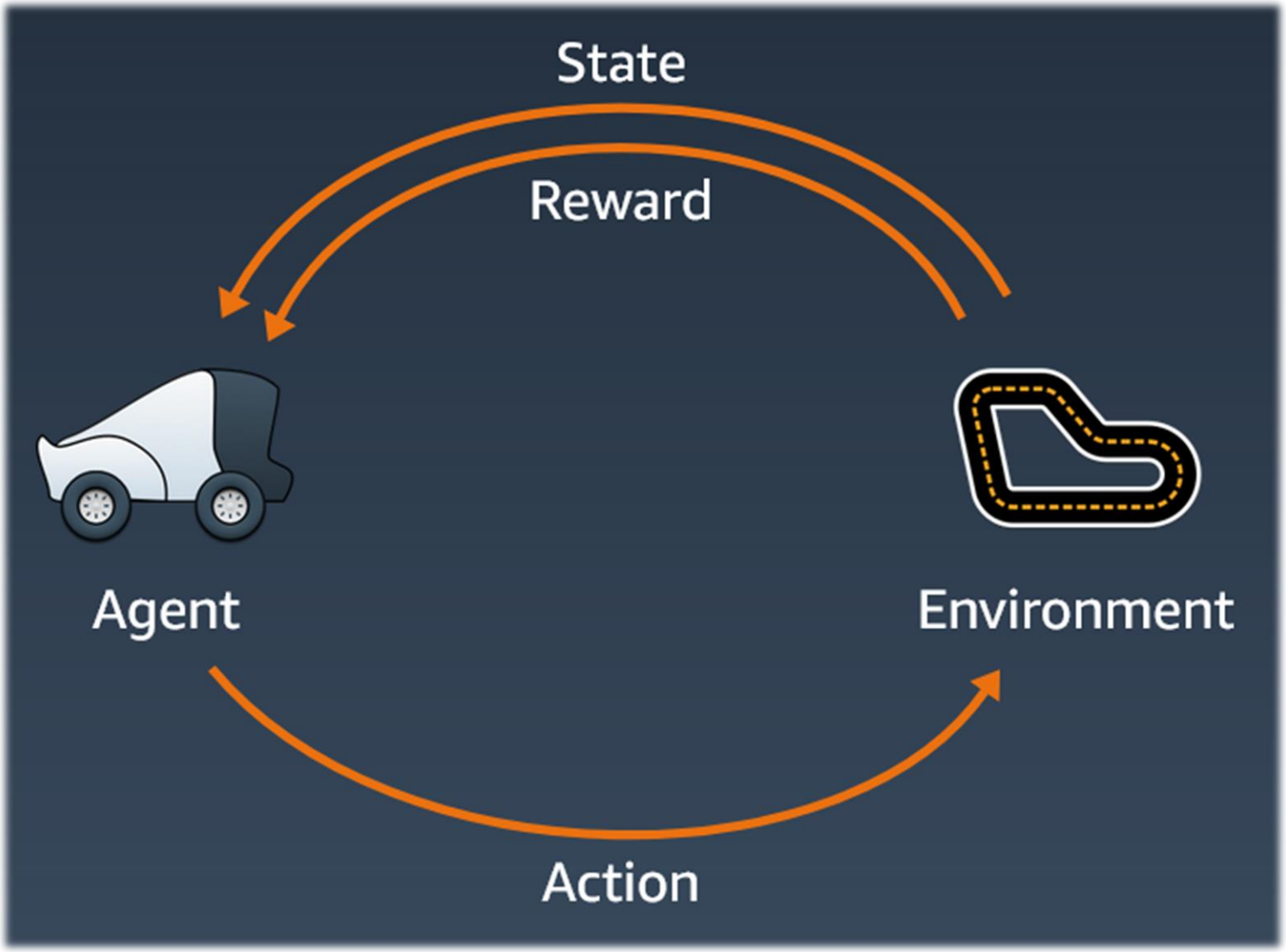
What is an AWS DeepRacer? (Under the Hood)



Inside of AWS DeepRacer:

- Intel Atom® Processor as a CPU,
- Ubuntu OS-16.04 LTS as OS,
- 7.4V/1100mAh lithium polymer as a car battery,
- 13600mAh USB-C PD as computer battery.
- 4 MP camera with MJPEG,
- 4GB RAM,
- 32GB expandable memory,
- 802.11ac Wi-Fi,
- IMU sensors (Integrated accelerometer and gyroscope)
- Intel OpenVINO toolkit
- ROS Kinetic

Reinforcement Learning (RL)

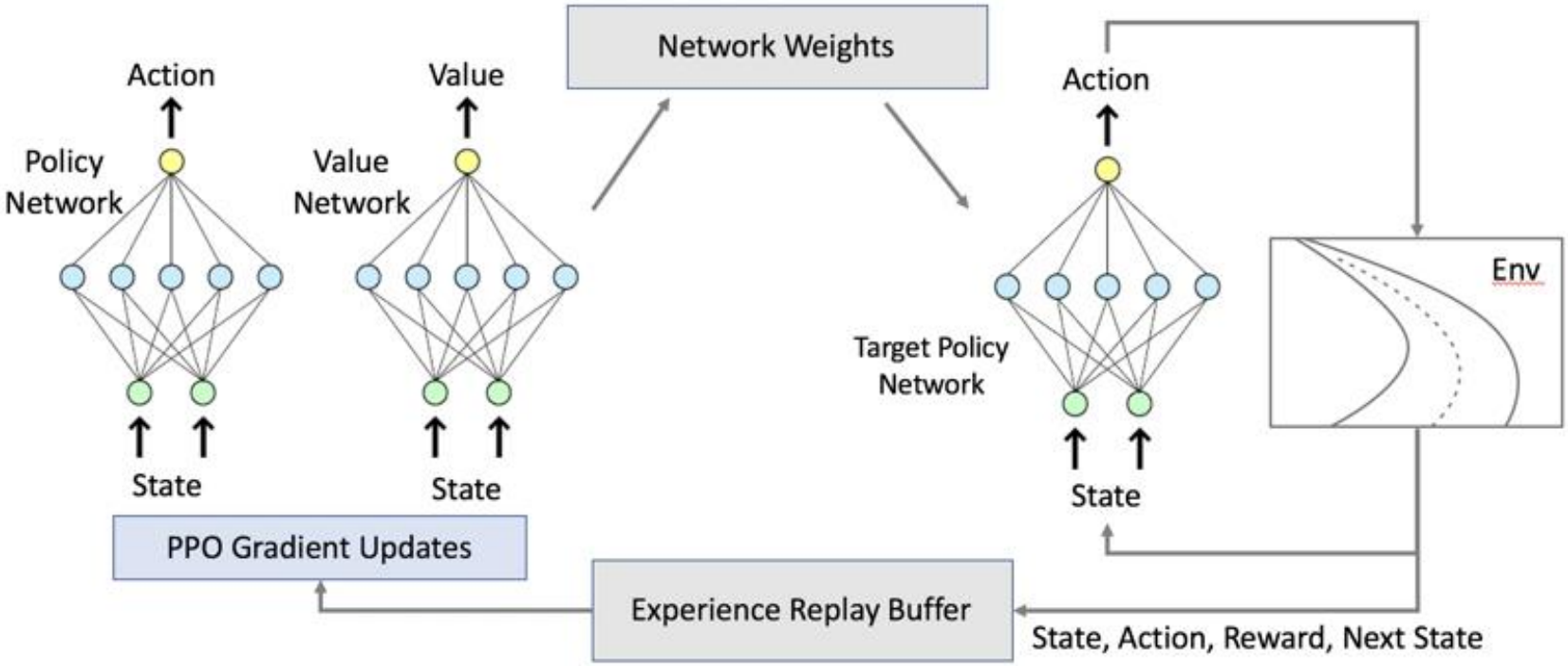


Reinforcement Learning (RL)

Components	Role
Agent	The *agent* simulates the AWS DeepRacer vehicle in the simulation for training. More specifically, it embodies the neural network that controls the vehicle, taking inputs and deciding actions
Environment	The *environment* contains a track that defines where the agent can go and what state it can be in. The agent explores the envrionment to collect data to train the underlying neural network
State	For AWS DeepRacer, a *state* is an image captured by the front-facing camera on the vehicle and/or the distance data by the LiDAR
Action	An *action* is a move made by the agent in the current state. For AWS DeepRacer, an action corresponds to a move at a particular speed and steering angle
Reward	The *reward* is the score given as feedback to the agent when it takes an action in a given state. In training the AWS DeepRacer model, the reward is returned by a *reward function*.

Training Algorithm

AWS DeepRacer uses the **Proximal Policy Optimization (PPO) algorithm** to train the reinforcement learning model



Uses **two neural networks** during training: **a policy network and a value network**

Training Algorithm

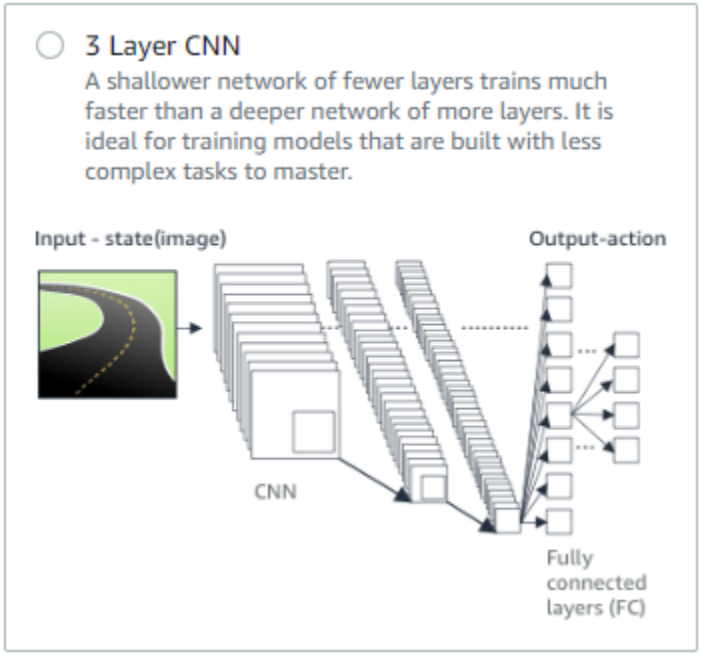
- **PPO is a derivative of the policy gradient method**, which trains the agent to move along a track by searching for the optimal policy
- The **policy network** (aka actor network) **decides which action** to take given an image as input
- The **value network** (aka critic network) **estimates the cumulative reward** we are likely to get given the image as input

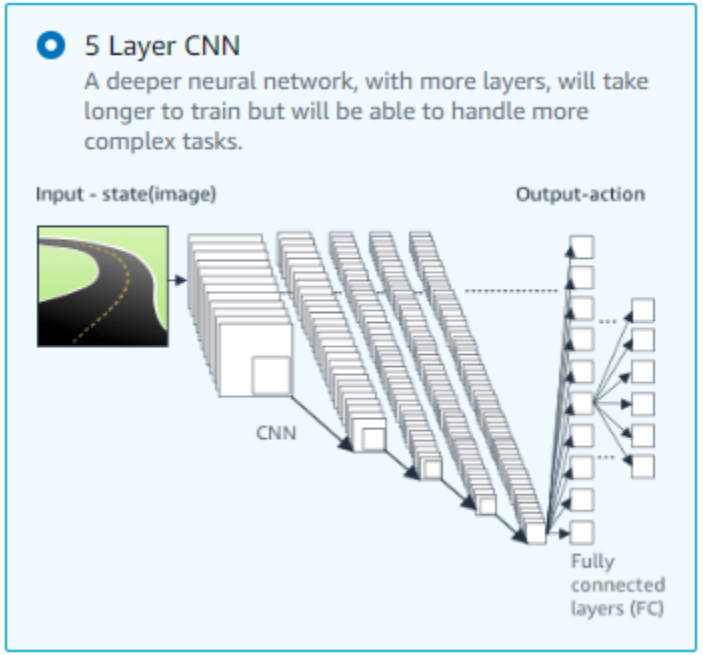
Neural Network Topologies

- The neural network is the **core of your machine learning model** that **processes sensory inputs into actions** for your vehicle
- The depth of the neural network defines the complexity of the model and it's ability to perform tasks
- A **deeper network can learn more complex behaviors** (e.g. sharp curves, numerous turns, avoiding obstacles) **than a shallow network** (i.e. more vs fewer layers)

Neural Network Topologies

- AWS DeepRacer supports **3-layer CNN or 5-layer CNN** (New or Removed?)





Registering AWS Educate Student Account (Important!)

- Go to the following link: https://www.awseducate.com/registration#APP_TYPE
- Click on “Student”
- Under School Name, type and select from the list: “Institute of Systems Science, National University of Singapore”; it will auto appear as you type this
- Under Email, provide a valid, current email issued by NUS ISS; e.g. your_name@u.nus.edu
- Fill in the rest of the details accordingly and click “Next”
- Scroll through the entire T&C, click “I Agree” and click “SUBMIT”
- You have to wait a few working days for AWS Educate to approve your account; will be sent to your email once approved

Instructions

- Login into AWS Educate using the account that you register previously
- Go to “Classrooms & Credits”
- Under “Classrooms where I am a Student”, under “Status”, click “Go to classroom”
- You will be directed to your workbench. This is where you monitor the amount of credits you have left. Click on “AWS Console”
- In your AWS Console, go to the search bar at the top and type “AWS DeepRacer”. Click on it to be directed to the AWS DeepRacer platform

Getting Started (Optional)

- At the side bar, click “Get started”
- Complete “Step 1: Learn the basics of reinforcement learning” (or go into this link: https://d2k9g1efyej86q.cloudfront.net/)
- Complete “Step 3: Learn about sensors and new types of racing” (or go into this link: https://docs.aws.amazon.com/deepracer/latest/developerguide/deepracer-choose-race-type.html?icmpid=docs_deepracer_console)

*Note that most of what are covered in these steps are already covered in the previous modules or will be covered in the next few slides

Creating a RL Model

- At the side bar, click “Your models”; this is where all your created/trained models will be stored
- Click “Create model”
- Put in a suitable model name
- For the environment, choose “re:Invent 2018”; this is recommended for beginners, hence will use for this course. You can try out other maps at your own time
- Click “Next”

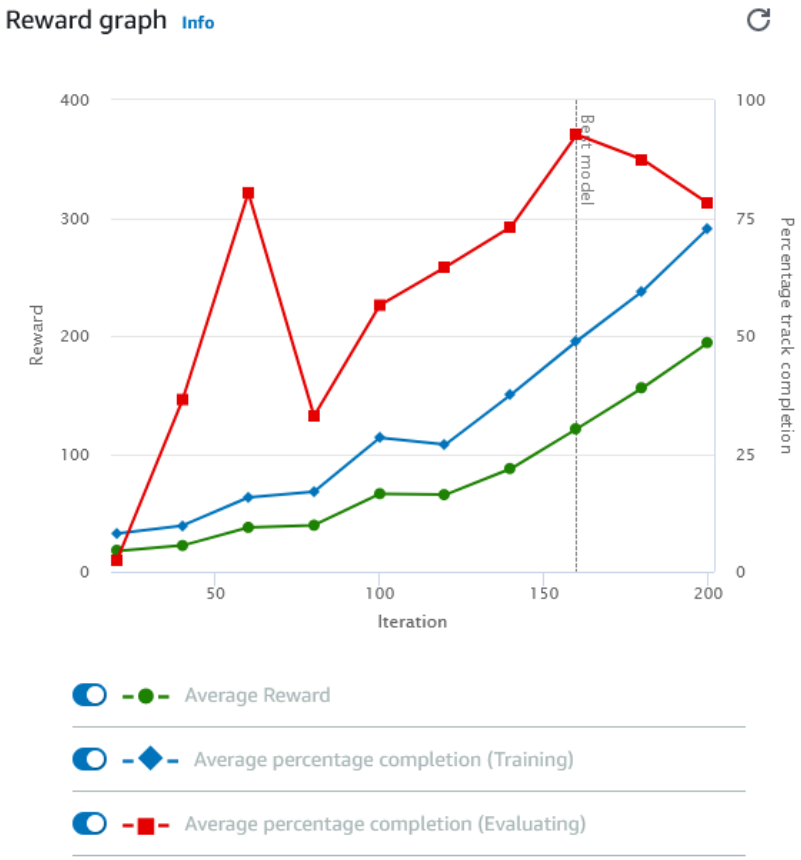
Creating a RL Model (Cont)

- Choose race type: “Time trial”; again, this race type is recommended for beginners. You can try more complex race type like “Object Avoidance” at your own time
- Under Agent, choose “The Original DeepRacer”; this is the default car model. You can customize and use other car models of your preference for your races (to be covered later on)
- Click “Next”
- Under Reward function, click on “Reward function examples”

Creating a RL Model (Cont)

- Expand “Time trial - follow the center line (Default)” and click on “Use code”; we will use this simple reward template for the very first training run; you are required to explore trying out other templates and/or editing the original template later on for your Try-Outs
- Under Training algorithm and hyperparameters, leave all settings to its default; you can adjust them later on for your Try-Outs
- Under Stop conditions, set “Maximum time” to 60 mins
- Check the box to “Submit the model to the DeepRacer League automatically” and click “Create model”

Training your RL Model



- Give it time to initialize and to begin training
- When it starts training, the Reward graph (Example on the Left) will be updated over time
- You can view the simulation video stream while it trains
- It is not necessary to leave the window open for the training to continue; you can choose to close it

It is ideal that the best model coincides with a high amount of percentage completion (for Evaluating; Red line)

Parameters that can be used

In total there are 13 parameters you can use in your reward function

x and y	The position of the vehicle on the track
heading	Orientation of the vehicle on the track
waypoints	List of waypoint coordinates
closest_waypoints	Index of the two closest waypoints to the vehicle
progress	Percentage of track completed
steps	Number of steps completed
track_width	Width of the track
distance_from_center	Distance from track center line
is_left_of_center	Whether the vehicle is to the left of the center line
all_wheels_on_track	Is the vehicle completely within the track boundary?
speed	Observed speed of the vehicle
steering_angle	Steering angle of the front wheels

If you need visualization, go to the link below; click “Next” until **“Parameters of reward functions”**:

https://d2k9g1efyej86q.cloudfront.net/

Available Reward Templates

You can view them under “Reward function examples” while creating the RL model:

1. **Time trial - follow the center line (Default)**
2. **Time trial - stay inside the two borders**
3. Time trial - prevent zig-zag
4. **Object avoidance** and head-to-head - stay on one lane and not crashing (default for OA and h2h)

Evaluating your RL Model

- Once training is completed, you can start evaluating your model on the same page by clicking “Start new evaluation”
- Select the environment that you train the RL Model in (i.e. re:Invent 2018)
- Scroll down and select 5 trials instead of the default 3
- Leave Race type as “Time trial” and click “Start evaluation”

Evaluating your RL Model (Cont)

- Give it time to initialize and to begin evaluation
- When it starts training, the Evaluation results (Examples on next slide) will be updated for each trial
- You can view the simulation video stream during the evaluation
- It is not necessary to leave the window open for the evaluation to continue; you can choose to close it

Evaluating your RL Model (Cont)

Evaluation Results Examples:

Trial	Time	Trial results (% track completed)	Status
1	00:00:28.269	100%	Lap complete
2	00:00:27.818	100%	Lap complete
3	00:00:25.687	100%	Lap complete

Trial	Time	Trial results (% track completed)	Status
1	00:00:21.380	100%	Lap complete
2	00:00:20.911	100%	Lap complete
3	00:00:20.871	100%	Lap complete

Changing Vehicle Model

- At the sidebar, click “Your garage”; you will see the default vehicle here
- Click on “Build new vehicle”
- Choose the type of camera you want (RGB or Stereo), and whether you want LIDAR sensor. When done, click “Next”
- You can choose to alter the details in the “Action space”. E.g. the maximum speed and maximum steering angle. When done, click “Next”
- Name your customized vehicle and choose the desired colour for it. Click “Done” after which

Changing Vehicle Model (Cont)

Examples:

RacerWithLidar | Mod vehicle

Sensor(s)

Camera
Lidar

Neural network topology

3 Layer CNN

Action space

Speed: 1 m/s Steering angle: 30°

Shell type

DeepRacer

Changing Vehicle Model (Cont)

Examples:

Racer_StereocameraWithLidar Mod vehicle

Sensor(s)

Stereo camera
Lidar

Neural network topology

3 Layer CNN

Action space

Speed: 1 m/s Steering angle: 30°

Shell type

DeepRacer

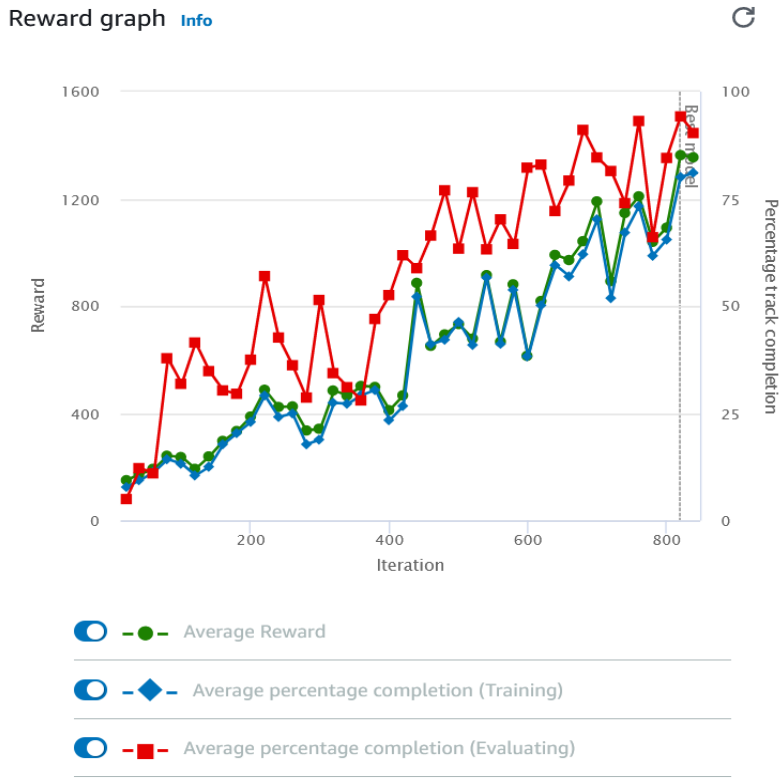
Today's Try-Outs

Using **"re:Invent 2018"** environment and **WITHOUT changing the "Action space" of the vehicles** (i.e. the default values),

1. Train a model that enables **a Racer with RGB camera and LiDAR** to complete the race
2. Train a model that enables **a Racer with ONLY Stereo camera** to complete the race
3. Train a model that enables **a Racer with Stereo camera and LiDAR** to complete the race
4. Train a model that is able to **beat all your previous time records**, and able to complete the race. **Hint: you have to alter the Reward Function**

Submission Instructions

Submit a print screen of your best model's reward graph and evaluation results in the "AWS DeepRacer Print Screen Results" submission folder; for example:



Evaluation results

Trial	Time	Trial results (% track completed)	Status
1	00:00:25.090	100%	Lap complete
2	00:00:24.474	100%	Lap complete
3	00:00:22.441	100%	Lap complete
4	00:00:22.867	100%	Lap complete
5	00:00:08.191	27%	Crashed

Optional Try-Outs

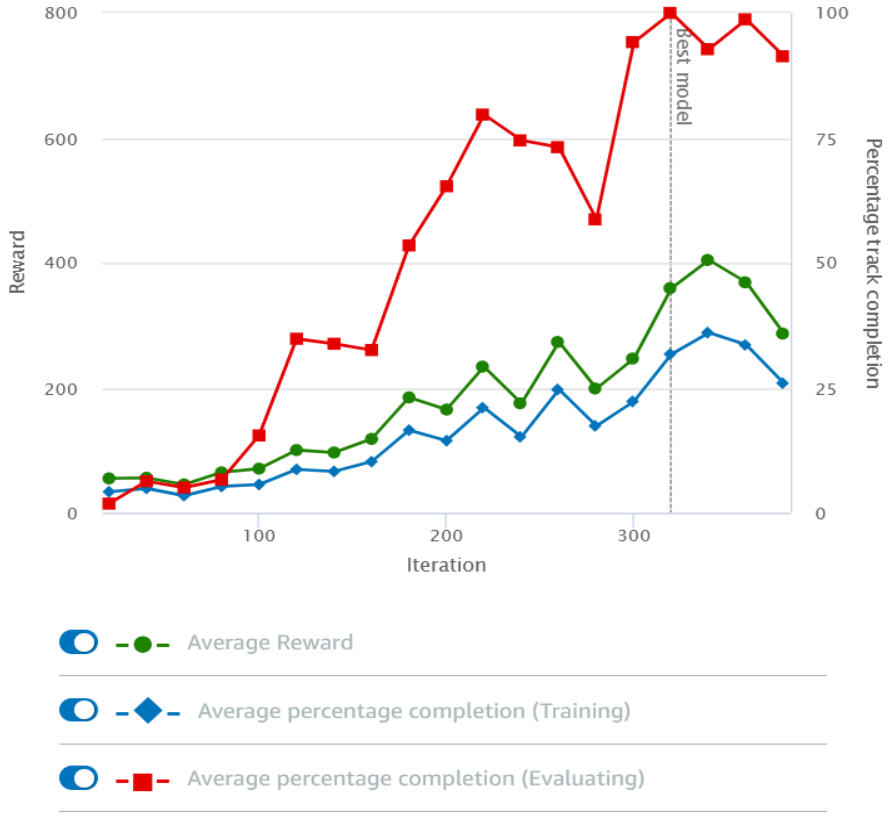
Below are some examples which you can try out yourself at your own free time; note that these are more challenging situations, which **may require more training time (3 to 4 hours)**

1. Using "**American Hills Speedway**" environment, train a model that enables a Racer to complete the race; you can choose any Racer you want
2. Using "**re:Invent 2018 Wide**" environment, select **Race type "Object avoidance"**, and train a model that enables a Racer to complete the race without crashing into any obstacles; you can choose any Racer you want

Optional Try-Outs

Training & Evaluation Results for No. 1 (i.e. American Hills Speedway):

Reward graph Info



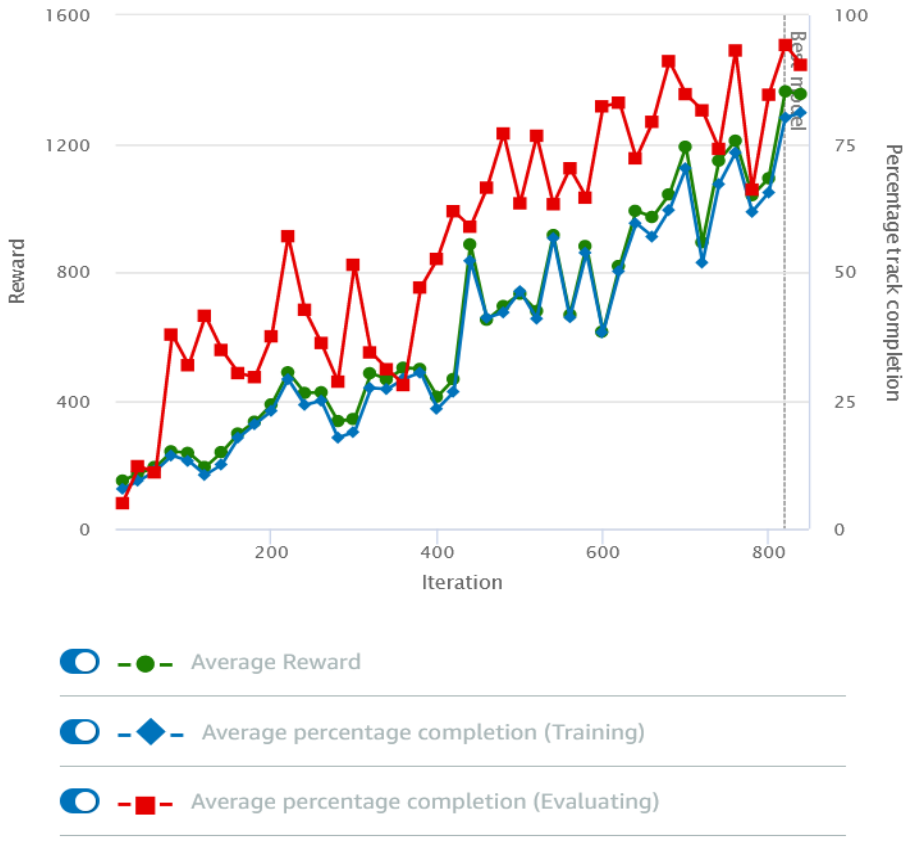
Evaluation results

Trial	Time	Trial results (% track completed)	Status
1	00:00:54.908	78%	Off track
2	00:01:09.815	100%	Lap complete
3	00:01:08.584	100%	Lap complete
4	00:01:08.269	100%	Lap complete
5	00:01:07.230	100%	Lap complete

Optional Try-Outs

Training & Evaluation Results for No. 2 (i.e. re:Invent 2018 Wide; Obstacles):

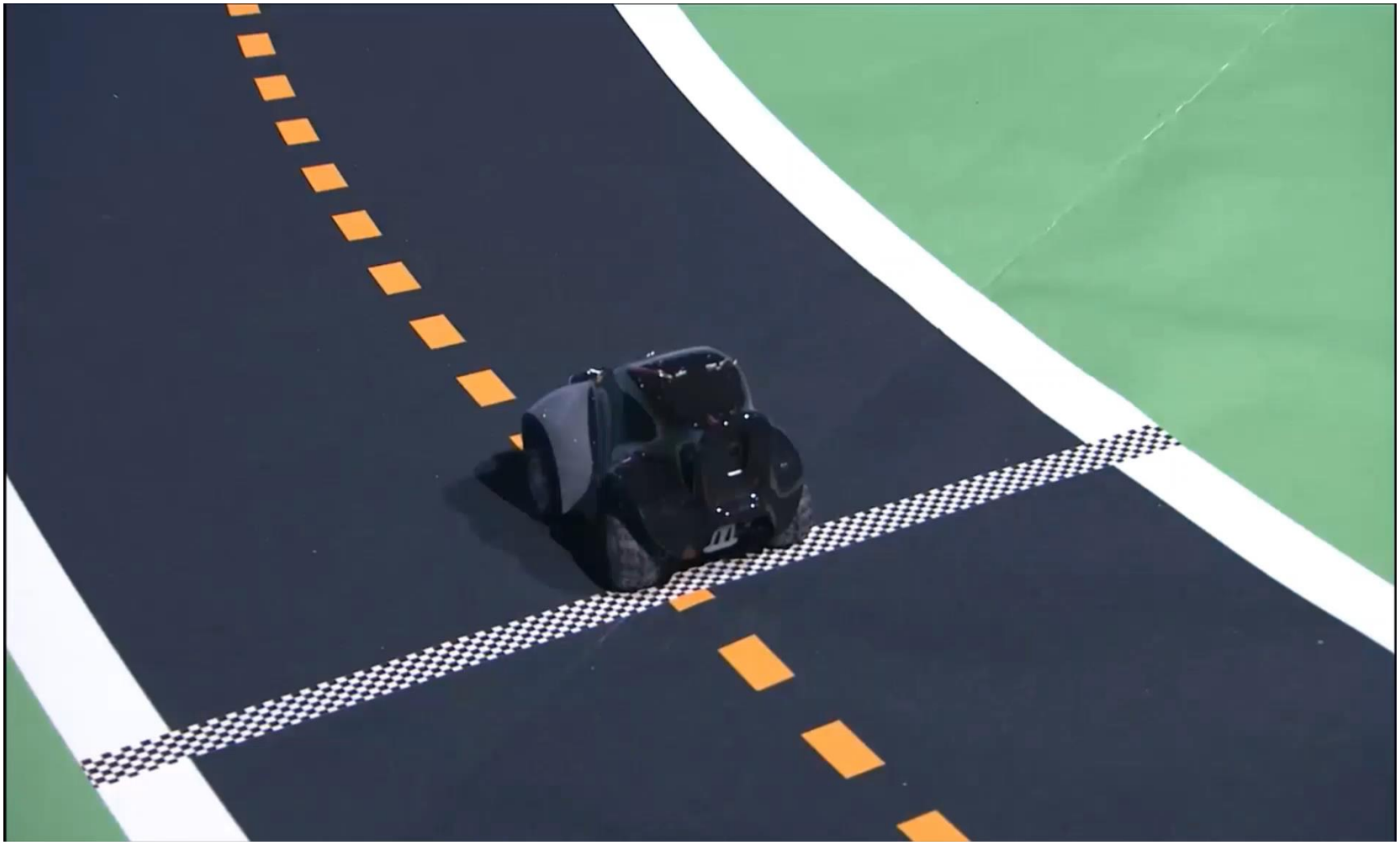
Reward graph Info



Evaluation results

Trial	Time	Trial results (% track completed)	Status
1	00:00:25.090	100%	Lap complete
2	00:00:24.474	100%	Lap complete
3	00:00:22.441	100%	Lap complete
4	00:00:22.867	100%	Lap complete
5	00:00:08.191	27%	Crashed

The Real Deal



Source: https://youtu.be/xL34jRhg6ME?t=427

THANK YOU

Email: nicholas.ho@nus.edu.sg