

Model variables for AWS Deep Racer

Introduction:

Reinforcement learning (RL) intends to solve the problem of reaching goals when the combination of possible feature combinations is of unmanageable proportions. For DeepRacer (DR) this means that getting around the racetrack in quickest time depends on too many factors under the agent racer's and data scientist's control. I have listed the principal controllable features below.

The general idea here is that if you know RL better than your competitor and you are willing to spend more money on training time, you can get a DR around the track quicker than a competitor. RL comprehension determines modeling choices. Modeling choices determine dollars or time spent training. All will determine average lap time and thus position on leaderboard. So, I believe that worthy metrics for model comparison will include:

- average lap time
- \$\$ or time spent training
- settings for the following model features

Modeling choices:

- training time = \$\$
 - generalization vs overfitting
 - poor vs excellent state x action space combination comprehension
 - train set vs validation vs simulated test sets vs physical test set
- car sensors
 - Face forward cameras
 - Mono
 - Stereo (depth perception)
 - LIDAR (object avoidance)
- training algorithm (one only = actor critic)
- deep learning network
 - 3 or 5 layers
 - 7 hyperparameters
- Action space
 - Race type
 - Time trial (TT)
 - Obstacle avoidance (OA)
 - Random vs fixed
 - Number of obstacles
 - Head to head (H2H)
 - Agent can change lanes or cannot
 - Competitors can change lanes
 - Random intervals
 - Fixed intervals
 - 1 to 4 competitor bots

- Action space (continued)
 - Car set up
 - Speed
 - Max
 - Number of speed increments
 - Turning angle
 - Max
 - Number of angle increments
 - Track type
 - Number and arc of turns
 - Mixture of straights
- Reward function
 - Complexity
 - 20 parameters
 - For example: speed, angle, position, off_track, is_crashed, distance_from_center
 - 4 sample functions to start with / can mix these
 - follow center line
 - stay inside borders
 - prevent zigzag
 - stay in one lane without crashing
- Model cloning (transfer learning / evolutionary learning)
- Validation
 - Time validating
 - Similarity to
 - Training
 - Simulated test
 - Physical test
- Framework
 - Sagemaker
 - Tensorflow
 - Other
- Services consumed
 - Direct vs indirect by AWS DR
 - List
 - AWS S3 = simple storage service
 - AWS Lambda
 - AWS CW
 - AWS CW logs
 - AWS Cloud formation
 - Sagemaker
 - AWS Robomaker
 - Kinesis video
 - application autoscaling
 - EC2
 - ECR
 - Kinesis video

Next steps:

- IAM roles
- Logs <https://github.com/aws-samples/aws-deepracer-workshops/tree/master/log-analysis>

