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 data scientist modeler's control. After a review of DR Developer Guide, I have listed a modeler's 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
- lap completion rate
- $$ 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
        - Gradient descent batch size
        - Number of epochs
        - Learning rate
        - Entropy
        - Discount factor
        - Loss type
        - Number of experience episode per policy update iteration
- Action space
    - Race type
        - Time trial (TT)
        - Obstacle avoidance (OA)
            - Random vs fixed
            - Number of obstacles
        - Head to head (H2H)
            - Constant speed: Set speed > competitor bots to enable passing action

      - Agent can change lanes or cannot (2 lanes)
      - Competitors can change lanes at
        - Random intervals
        - Fixed intervals
      - 1 to 4 competitor bots
  - Car set up
    - Speed
      - Max
      - Numbe r 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
    - all_wheels_on_track
    - closest_waypoints
    - closest_objects
    - distance_from_center
    - heading
    - is_crashed
    - is_left_of_center
    - is_offtrack
    - objects_distance
    - objects_left_of_center
    - objects_location
    - objects_speed
    - progress
    - speed
    - steering_angle
    - steps
    - track_length
    - track_width
    - waypoints4
  - 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)
  - Does the model employ transfer learning?
- Validation
  - Time validating
  - Similarity to
    - Training set
    - Simulated test set
    - Physical test set

- Framework
    - Sagemaker
    - Tensorflow
    - Other
- Services consumed by AWS DeepRacer (DR)
    - List
        - AWS S3 (Simple Storage Service)
        - AWS Lambda
        - AWS CloudWatch (CW)
        - AWS CW logs
        - AWS CloudFormation
        - Amazon Sagemaker
        - AWS RoboMaker
        - Amazon Kinesis Video Streams
        - Application Auto Scaling
        - EC2 (Elastic Compute Cloud)
        - Amazon Elastic Container Registry (ECR)
    - Resources directly called by AWS DR
        - AWS S3
        - AWS Lambda
        - AWS CW
        - AWS CW logs
        - AWS Cloud formation
        - Amazon SageMaker
        - AWS RoboMaker
        - Amazon Kinesis Video Streams
    - Resources indirectly called by AWS DR (and callers)
        - AWS S3
            - AWS Robomaker
            - Amazon SageMaker
        - Amazon CW
            - AWS Robomaker
            - Amazon SageMaker
        - Amazon CW logs
            - AWS Lambda
            - AWS Robomaker
        - Application Auto Scaling
            - Amazon Sagemaker
        - EC2
            - Amazon SageMaker
            - AWS CloudFormation
        - ECR
            - AWS RoboMaker
        - Amazon Kinesis Video Streams
            - AWS Robomaker

Next steps:

- Definitions
- Racer agent configuration is recorded in model metadata; what else is recorded?
- Strategies to improve models: Reward functions, systematic hyperparameter tuning.
- Data protection/encryption in transit.
- IAM roles
- Logs https://github.com/aws-samples/aws-deepracer-workshops/tree/master/log-analysis