Hands-on Exercise 2: Advanced Model Training Using AWS DeepRacer Console

Overview

This exercise helps you put into practice what you learned in the first four chapters of this course on AWS DeepRacer and reinforcement learning. The exercise walks you through how to optimize your model by using an advanced version of the reward function and fine-tuning your model's hyperparameters.

*Note: This exercise is designed to be completed in your AWS account. AWS DeepRacer is part of AWS Free Tier, so you can get started with the service at no cost. For the first month after sign-up, you are offered a monthly free tier of 10 hours of Amazon SageMaker training, and 60 simulation units of Amazon RoboMaker (enough to cover 10 hours of training). If you go beyond those free tier limits, you will accrue additional costs. For more information see the AWS DeepRacer Pricing page.

Learning Objectives

By the end of this exercise, you will be able to:

1. Clone a model in AWS DeepRacer
2. Configure a model by using the advanced reward function in AWS DeepRacer
3. Customize a model's reward function using parameters
4. Train and evaluate a model using the AWS DeepRacer simulator
5. Analyze how setting various hyperparameters can impact a model's performance

Technical Prerequisites

- Experience using AWS technologies
- Basic understanding of machine learning concepts, particularly reinforcement learning and how it applies to AWS DeepRacer
- A default model created in the AWS DeepRacer console (see Chapter 2 for instructions)

Task 1: Model tuning with the advanced reward function

In chapter 2, you created your first trained model using the default settings in console. If you do not have a trained model yet, see Chapter 2 to create and train your first model so that you can establish a benchmark for this exercise. This is a prerequisite for this exercise.

In this task, you will improve the simulation time you recorded with the default model you built in Chapter 2. That model was built using the basic reward function in the AWS DeepRacer console. Now you'll build on that model by using the advanced version of the reward function.

To set up your next model, follow these steps.

1. In your AWS account, go to the AWS Management Console.
2. Choose `us-east-1` region at the top right corner of the **Regions** dropdown menu.
3. From the top left of the console, click **Services**, type **DeepRacer** in the search box, and select **AWS DeepRacer**. That will open the AWS DeepRacer console.
4. From the AWS DeepRacer console, click **Reinforcement Learning**.
5. Select the model you built in Chapter 2.
6. From the **Action** dropdown, select **Clone**.
7. Name the model and add a description.
8. In the **Environment simulation** section, select the same track used in your first model.
9. Under **Action space**, familiarize yourself with these settings and accept the defaults.
10. In the **Reward function** section, select the **Advanced function** dropdown arrow and then select the first **Insert code** button. This code block optimizes for turning and requires less time to train. You can also scroll down a little to select the second **Insert code** button. The code for this function focuses on optimizing for speed. It also requires less training time.
11. In the **Algorithm settings** section, accept the default settings.
12. For **Stop conditions**, choose a max time of 60 minutes and click **Start training**. The new training will initiate in 5 to 6 minutes. You then must wait for the training job to complete before proceeding. If you choose a max time of 60 minutes, it will take up to 60 minutes for this training job to complete. It is complete when the status reads Ready.

Tip: One effective way to train your model is to use shorter training session followed by adjustments and then more training. You may want to compare this type of approach to longer training jobs exceeding several hours.

Note: While the training job is running, we encourage you to learn more about AWS DeepRacer's reward function by reading through some relevant information in the AWS DeepRacer documentation. You can also get started on Chapter 5 of this course during this time.

13. Now, select the name of the model you just trained.
14. Click **Start new evaluation**.
15. Select the track (this should be the same as the track you trained on).
16. Set the number of trials to **3** and click **Start evaluation**. The evaluation results will update in approximately 5 minutes.

    Note: While the evaluation job is running, watch the live results in the simulator. What's your car doing that seems to be working well? Is it staying on the center line? Is it moving at a good speed? What's your car doing that isn't working well? How's it doing around corners, for instance?
17. Compare the results from your evaluation with the model you built in Chapter 2. How do they compare? What factors contributed to differences in time or performance?
18. After watching your evaluation job in the simulator and comparing your evaluation results to your original performance, consider repeating the steps in Task 1—but this time, experiment with different parameters and rewards in your reward function. See the Tips and Tricks section below for some suggestions on how to get started.

Tips and tricks

- Try to increase your speed setting and stabilize the training.
- Use progress and speed variables to reward higher speeds and faster lap times.
- Keep as close to the center line as possible.

Task 2: Model tuning with different hyperparameter settings

In Task 2, you'll make changes to your model's hyperparameters. You'll first clone your existing model, adjust the hyperparameters, and then train and evaluate the new model.

1. From the AWS DeepRacer console, click **Reinforcement learning**.
2. Select the model you want to optimize (for the purposes of this exercise, select the model you used in Task 1).
3. From the **Action** dropdown, select **Clone**.
4. Name the model and add a description.
5. Select the same track you used in Task 1.
6. Under **Action space**, keep the default settings.
7. Keep the **Advanced reward function** checked.
8. Experiment with changing one or two hyperparameters at a time to see how those changes impact your model (for instance, adjust entropy, learning rate, epochs, or another hyperparameter).
9. For the **Stop condition**, set the max time at **60** minutes and click **Start training**.
10. Select the name of the model. Then watch the live stream in the simulator, notice how your car is moving, and get familiar with the look and feel of the simulation. The new training will initiate in about 6 minutes. You then must wait for the training job to complete before proceeding. If you choose a maximum time of 60 minutes, it will take up to 60 minutes for this training job to complete. It is complete when the status reads Ready.

    Note: While the training job is running, we encourage you to learn more about AWS DeepRacer's hyperparameters by reading through some relevant documentation. You can also get started on Chapter 5 of this course.
11. Next, select the name of the new model you just trained and start a new evaluation job.
12. Select the same track you just trained on.
13. Set the number of trials to **3** and click **Start evaluation**. The evaluation results will update in approximately 5 minutes.

    Note: While the evaluation job is running, watch the live results in the simulator. Remember to look at what your car seems to be doing well versus not so well.
12. Compare the results from your evaluation with your previous two performances. How did they compare? Did the lap time improve with different hyperparameter settings? If so, why do you think that happened? If your lap time didn't improve, what do you think you could have done differently to improve performance?
13. Consider repeating Task 2 by adjusting other hyperparameters and comparing your results. See the Tips and Tricks section below to get started.

Tips and Tricks

- One way to improve your model's performance is to enact a better or more effective training process. For example, to obtain a robust model, training must provide your agent with more or less evenly distributed sampling over the agent's action space. This requires a sufficient mix of explorations and exploitations. Variables that affect this include the amount of training data used (number of episodes between each training and batch size), how fast the agent can learn (learning rate), and the portion of exploration (entropy).
- To make training practical, you may want to speed up the learning process, or slow it down. Variables that affect this include learning rate, batch size, number of epochs, and discount factor.

Conclusion

Congratulations! You now have successfully:

1. Cloned a model in AWS DeepRacer
2. Configured a model using the advanced reward function in AWS DeepRacer
3. Customized a model's reward function by experimenting with different parameters
4. Trained and evaluated a model using the AWS DeepRacer simulator
5. Analyzed how setting various hyperparameters can impact a model's performance

---

Additional Resources

- For more information about AWS DeepRacer, see https://aws.amazon.com/deepracer/.
- For more information about AWS Training and Certification, see https://aws.amazon.com/training/.
- To troubleshoot and collaborate about the AWS DeepRacer, see https://forums.aws.amazon.com/forum.jspa?forumID=318.

For feedback, suggestions, or corrections, email us at aws-course-feedback@amazon.com.