The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
Portland, OR . June 23 - 26, 2024
Paper ID #43969

Incorporating Artificial Intelligence into Mechanical Engineering with Amazon DeepRacer

Dr. Pooya Niksiar, The Citadel

Dr. Niksiar is assistant professor of Mechanical Engineering Department at The Citadel. He received his Ph.D. in Mechanical Engineering from Clemson University, his M.Sc. from K. N. Toosi University of Technology and his B.Sc. from Isfahan University of Technology, Iran. Prior to joining The Citadel, he was a lecturer at Clemson University. His research includes the design and development of advanced functional porous materials for bio applications. He has published several peer-reviewed journal papers in the area of his research, has received grant form NASA and Amazon has been reviewer for several journals. Dr. Niksiar is teaching graduate and undergraduate courses in the area of thermal and fluid Sciences, aerodynamics, materials, design, measurements and numerical methods.

Blakeley Hunter Odom, The Citadel

Incorporating Artificial Intelligence into Mechanical Engineering with Amazon DeepRacer

Pooya Niksiar, Blakeley Odom
Mechanical Engineering Department, The Citadel, Charleston, SC

Abstract

Artificial Intelligence (AI) is impacting the world similarly to how Industrial Revolution and Digital Revolution impacted the world in 18^{th} and 20^{th} centuries. The influence of Artificial Intelligence in shaping the future is inevitable and crucial for students in any major to acquire the skills needed to utilize AI in their respective fields and careers. One of the most effective approaches to introducing a new topic is by involving students in competitions. Amazon DeepRacer offers an excellent opportunity to introduce Machine Learning and Artificial Intelligence to the student body, providing essential tools and training to get started. In this study, a group of Mechanical Engineering students at The Citadel formed the artificial intelligence (AI) club and trained an Amazon DeepRacer car to follow a predefined trail. This study details the steps we took to train the car and compete in Amazon DeepRacer competition among senior military colleges.

Introduction

Hands-on activities are a key factor in effective engineering education. There are several ways to get students involved in the activities that they can apply knowledge learned in the classrooms to the real-world prototypes. Student clubs, competitions and projects are an excellent experience for students to design, build, test and troubleshoot real world functioning systems. Another great advantage, specifically for institutions focused on undergraduate teaching, is learning skills in conducting research to optimize, improve or add features to their prototype. Examples of engineering student clubs include rocketry club [1], Baja SAE club [2,3] and Robotic club [4]. In the past decade, the advent of Graphical Processing Units (GPUs) accelerated research and applications in the fields requiring intense computations. Machine and deep learning were the fields that benefited significantly from GPUs as they are computationally, very demanding.

Although machine learning and deep learning have been used for decades, ChatGPT was the first application to demonstrate the power and usefulness of Artificial Intelligence (AI) to a public audience. Since then, many fields have utilized AI to their advantage. The power and effectiveness of AI in many fields have led many to believe the next revolution like agriculture, the industrial revolution, and technology will be centered around Artificial Intelligence.
As it is crucial for students to equip themselves with skills in artificial intelligence to succeed in their future career or graduate studies, at Mechanical Engineering Department at The Citadel we formed AI club in summer 2023 and hosted the Amazon DeepRacer competition among senior military schools in November 2023. In summer 2023 the Citadel AI club started researching artificial intelligence to have general idea about the field and then focused specifically to Amazon DeepRacer on how that works. Meanwhile the adviser of the club, Dr. Niksiar, took two Amazon bootcamps on machine learning and deep learning to be able to guide the team with specific technical details. After the initial research on AI and DeepRacer in Summer 2023, Citadel team focused on choosing appropriate hyper-parameters and writing reward function to get ready for competition on November 8th. The emphasis here was on self-taught experience, meaning students take responsibility for learning and implementing DeepRacer with minimal guidance from adviser. At the end, this objective was met perfectly, and Citadel team did a great job on this. Amazon DeepRacer is a great way to introduce AI to the student body in any major. Amazon has created an infrastructure for students or universities who are interested in learning AI through Amazon DeepRacer. Amazon provides tutorials, documentation, and sample code to help developers get started. Amazon DeepRacer is supported through Amazon Web Services (AWS) in which developers can run their models on a cloud-based platform. Below we explain the structure and features of Amazon DeepRacer so that developers interested in using DeepRacer can follow along and start using DeepRacer. Although the perception is artificial intelligence is highly related to computer science, in this study we show that with the aid of Amazon DeepRacer infrastructure, all majors can get involved.

Amazon DeepRacer

Amazon DeepRacer is a 1/18 scale autonomous race car with a front camera receiving input data as an image (**Figure 1**, left). Amazon DeepRacer uses reinforcement learning, a subcategory of machine learning for its operation. Reinforcement learning works based on the reward concept, the car collects reward as long as it stays on track. In reinforcement learning, the car goes through many iterations which is called training and can range from 1-10 hrs. The training process is done in the simulator which will be discussed in the next section. AWS already has 61 different tracks for Amazon DeepRacer and we chose the “A to Z Speedway” (**Figure 1**, right) for this competition and the cars were running counterclockwise. Also, there are three different race types that can be chosen:

- Time Trial: performance is evaluated based on the fastest time
- Object avoidance: the car is racing on a two-lane track with a fixed number of stationary obstacles placed along the track
- Head-to-head racing: the car is racing against other cars on a two-lane track.

The time trial was chosen for the competition.

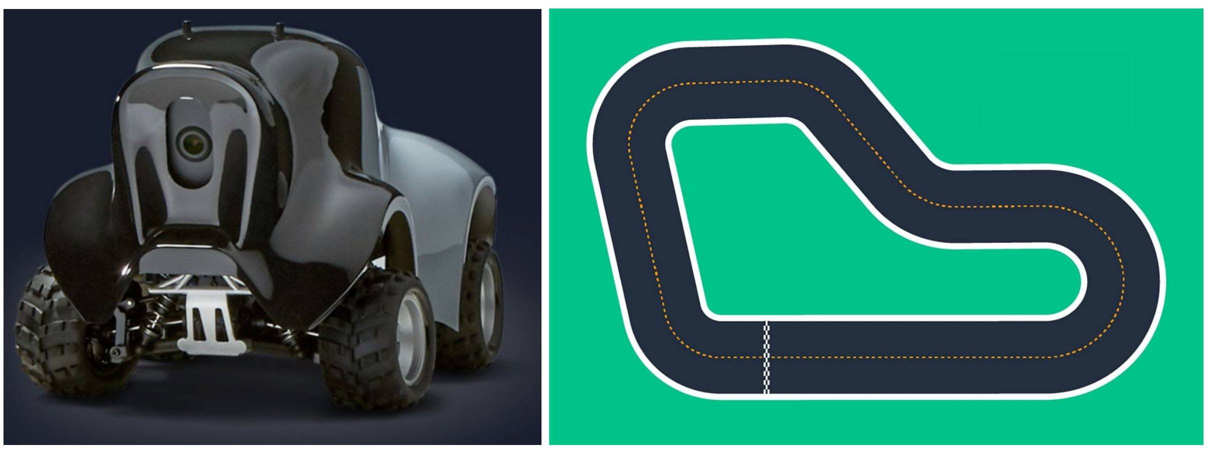

Figure 1. (left) Amazon Deep Racer car. **(Right)** simulator environment [5]

Simulator

The training process is done on a simulator which makes the process way more convenient and easier. This means that during the training process whenever the car gets off the track the user does not need to pick it up and put it on the start point again and repeat this for several hours of training. Instead, the developer can create the model in the simulator and let the car train itself. Whenever it gets off track, the computer puts it on the start point again for a new iteration of training. **Figure 2** shows a car going through training on the simulator. There are several factors that come into play regarding how a model will perform. They can be boiled down into three main categories: hyperparameters, reward function, and training. Below we go over them briefly.

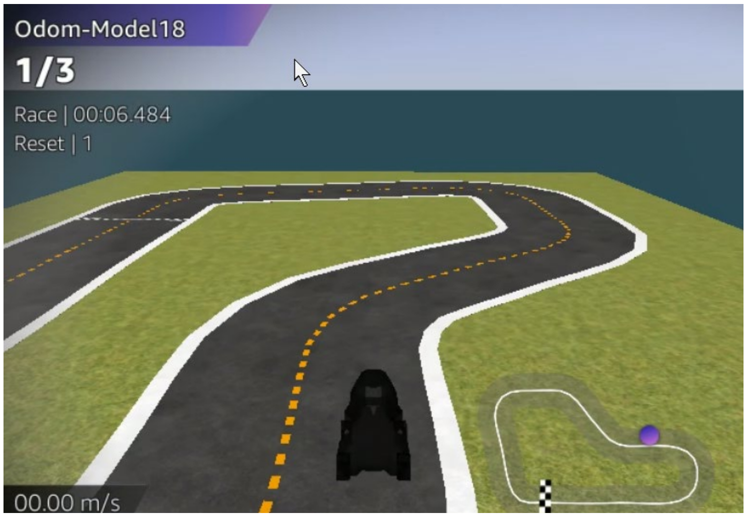

Figure 2. A to Z Speedway track was chosen for competition [5]

Reward Function

The reward function defines how the car is going to be rewarded for staying on track. As an example, **Figure 3** shows a reward function that rewards the car with 2 points if it stays on the center lane, 0.2 points if it enters black cells and zero points if it enters purple lanes. Through training, the car will learn that staying on the center line is the most beneficial behavior as it collects the maximum reward. Different rewarding mechanisms might be applied to keep the car on track depending on the purpose. As an example, one can be very conservative and select a very low speed for the car and penalize the car heavily in the case of it getting off the track to ensure that the car finishes a lap without getting off the track; this comes at expense of a long time run. In contrast, one might choose to pick a higher speed for the car and be less restrictive in regard to getting off the centerline and implement some mechanism to bring the car back to the center, this will result in a faster run for the car, provided it does not get off the track. Hence, with reward functions developers can be very creative and achieve a very competitive lap time with DeepRacer.

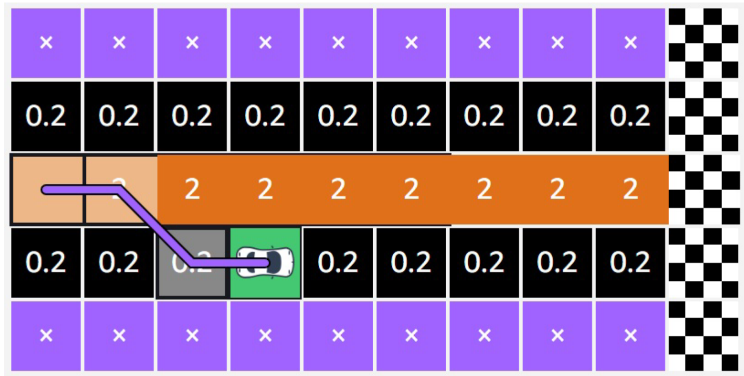

Figure 3. A to Z Speedway track was chosen for competition [5]

In **Figure 4,** we have shown an example of a reward function (this is not the code for reward function shown in **Figure 3**). Reward functions are written in python and utilize the information gathered from the car's internal camera to calculate the best position for it, as well as the best course to correct its position. The car is inherently aware of the information contained within the parameter matrix. This information includes the distance from the centerline of the track, the total width of the track, and the direction in which the wheels are currently turned. It uses this information to divide the track into 3 types of location; on target (orange lane in **Figure 3**), which is within a specified bound around the centerline, the midrange (black lane in **Figure 3**), when the car loses some of its reward but is not immediately penalized, and the outside of the track, (purple lane in **Figure 3**) where the car is directly penalized for being too far from the center, as the car is likely to end up off the track and requiring a reset. All these calculations are utilized to affect the variable "reward" which is then returned to the main code running within the car. The higher the value of the reward, the more likely the car is to repeat the actions it took previously.

```python
def reward_function(params):
    '''
    Example of penalize steering, which helps mitigate zig-zag behaviors
    '''
    # Read input parameters
    distance_from_center = params['distance_from_center']
    track_width = params['track_width']
    abs_steering = abs(params['steering_angle']) # Only need the absolute steering
angle

    # Calculate 3 marks that are farther and father 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

    # Steering penality threshold, change the number based on your action space
setting
    ABS_STEERING_THRESHOLD = 15

    # Penalize reward if the car is steering too much
    if abs_steering > ABS_STEERING_THRESHOLD:
        reward *= 0.8
    return float(reward)
```

Figure 4. Sample code for a reward function

Hyper-parameters

Hyper-parameters are the parameters that the user can change to optimize the model's performance for a specific application or dataset. Amazon DeepRacer has a total of 9 hyper-parameters that can affect the car's performance on the track, based on how well it stays on the track and how quickly it can finish a lap. Below we explain each hyper-parameter separately:

1- Gradient descent batch size. As the model trains, it gains experience, and stores it into an "experience bucket". The higher the value of gradient decent, the more experience the model uses for later iterations. A higher value results in a smoother operation, but it is more likely to learn more slowly or become overfitted. However, a gradient descent batch size that is too low would result in a model that cannot learn from past mistakes and successes at all [6].

2- Number of epochs- This hyperparameter is a measure of how often the neural network of the model updates. When this number is higher, each alteration of the network has more data to work with and therefore is going to be less random. When this value is small, the model may happen to stumble into success, and therefore it may train faster than a higher value; however, this will come at the cost of a model that is more sporadic and make seemingly nonsensical movements.
3- Learning rate- This value determines how much the model gains from gradient decent between each iteration. This number, being higher results in a model that will train much quicker but may not focus on a definite successful run.
4- Entropy- This value determines how much randomness will influence the network in each iteration. A higher number will result in a more potentially chaotic model, but it will find different routes that may result in a better or faster time. If this value is too high, the model does not learn from its actions, and it always moves randomly.
5- Discount factor- This value determines how many options the model have to choose from before each and every action, The larger this number is, the longer training takes, but the model may become more creative in determining its track lines
6- Loss type- This model can be one of two options, Mean Squared error, or Huber loss. Mean squared error tends to be a faster method but may be less likely to find a successful path.
7- Number of experience episodes between each policy-updating iteration- As previously mentioned, some of the prior hyperparameters refer to an experience bank. This model determines how large of a bank the model will have to be before making an alteration to the network, and as with many of these models. In reinforcement learning car will start with exploring the grid until it moves out of the boundary. As it moves around, collects rewards based on scores we defined for each location. This process is called an “episode”.
8- Steering angle- This parameter measures the steering of the car in degree, if the car is steering right, it is negative and if car is steering left, it is positive.
9- Speed- This parameter defines the speed of the car.

Competition

Competition was among three senior military schools, The Citadel, United States Naval Academy and United States Military Academy. The 2023 competition was hosted by The

Citadel. Three different departments participated from each school, Mechanical Engineering Department at The Citadel, Computer Science Department at Naval Academy and Electrical Engineering Department at Military Academy. A total of 10 teams participated, and each team had two runs. Each run lasted two minutes, and the car could go around the path as many times as it was able. If a car had finished several laps during its two minutes of time, then the best timing was considered for the lap. The first team was from The Citadel with timing of 10.73 s, the second team weas from The Citadel again with timing of 11.45 seconds and third team was from Military Academy with timing of 12.16 seconds.

Figure 5. Amazon DeepRacer car running on the track [7]

Performance

When training of the model is finished in the simulator, the results can be evaluated in the evaluation section. **Figure 6** shows the results of two sample models, the left one is a high performing model, and the right one is a low performing model. These graphs have three distinct lines. The green line represents the value of the reward variable and its progress as the model trains for more iterations. In high performing model (**Figure 6a)**, the reward value increases as the model trains for more time, while in low performing model (**Figure 6b**) the reward value fluctuates/decreases. Note that in **Figure 6b** the model obtains a high reward value (i.e. 200) in the first iteration is totally random. The red line represents the percentage of the track that the machine was able to complete during the evaluation runs after training. This is typically a measure of how well the machine will complete the track without veering off course. As can be seen from **Figure 6a** the model from iterations 17~27 has reached %100 completion which means if training is stopped at that point there is high probability that car will finish a lap without getting off the track. On contrast, the red line for the low performing model is decreasing which means the car will get off track further and further as model trains more. The blue line represents the progress of the model in terms of reduced run time or reduced off-track time through iterations, as can be seen from **Figure 6a** the progress has increased over iterations while for **Figure 6b** it fluctuates.

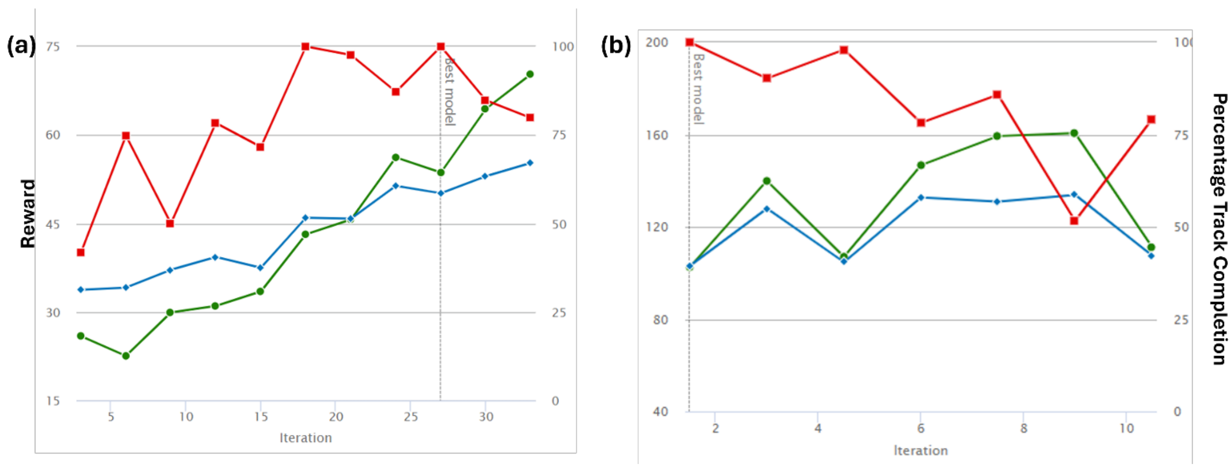

Figure 6. (a) is evaluation of a high performing model and **(b)** is evaluation of a low performing model . The red line shows the percentage of the track machine was able to complete in a run, the green line shows the total reward collected, and blue line shows success of car during training.

Discussion

The models created by teams were loaded on the Amazon DeepRacer car and were tested on the real track. Although the Amazon DeepRacer physical car and simulator are designed in the same way and work similarly, they are not identical and there were several difficulties in running the models on physical car. The camera of physical car was sensitive to the environmental light, making them easily veer off the track due to nuances from the surrounding environment. Generally, the models that were overfitted did not perform well on the physical car and tended to veer off the track frequently, while they might have good performance on the simulator. Overfitting occurs when a model learns the training data very well, picking up any noise and fluctuations in the data while missing the underlying pattern. An overfitted model will perform poorly on new or unseen data. On the other hand, underfitting happens when the model is too simple to capture the underlying structure of the data. Based on our experience overfitting must be avoided to make sure the car stays on the track.

Finally, the financial of competition was generously supported by amazon and there was $25 k grant for equipment, travel logistics, students training time and awards. With this grant we were able to purchase the track, car, router, barricades, reimburse students for the training hours they were using AWS to train their models and competition prizes. The first three teams with the best running time were awarded.

Student Perception

Citadel team members were asked about their perception about AI before and after the competition and how effective this competition was in familiarizing them with Artificial Intelligence. All students admitted that they got a better understanding of what AI is and how is used in different fields. Also, they mentioned after participating in the competition they realized AI is not purely related in computer science and can be used in many fields in mechanical engineering, they mentioned fields like, design, manufacturing, quality control and autopilot. They were also feeling nervous about it before the competition because they had little knowledge of AI and had never done such a thing. But curiosity and pushing them out of their comfort zone was the main reason for them to participate. Finally, the fact that different engineering majors participated in the competition made students more competitive and passionate about the competition, specifically for mechanical engineering students to prove themselves again computer science teams.

Conclusion

Amazon Deepracer competition was a great experience for undergraduate students to get involved and learn about machine learning by implementing that. Amazon has provided great infrastructure to facilitate students with limited knowledge of machine learning to participate. It is designed so that students can have access to a few reward functions and initial values of hyperparameters to get started, and when they enhance their skills, they can develop more complex models by adjusting reward function or hyper parameters. Providing tutorials and documentation has made it possible for all engineering majors to participate, with just basic programming knowledge required. This can be incorporated in ME curriculum or other engineering majors through student club competing with other universities or holding competition within the school among different departments and schools with different categories, for sophomore, junior and senior students. In fact, any engineering student who has passed a programming language course can participate.

References

[1] P. Niksiar, R. Integlia "Formation of the Citadel Aerospace and Rocketry Student Organization", *ASEE-SE*, Charleston SC, March 2022.

[2] M. Sadat-Hossieny et al., "Using SAE-Baja project in attracting and empowering women in engineering/technology: A case of success at Northen Kentucky University (NKU) and Cincinnati State Technical and Community College (CSTCC)", *Proceedings of the 2020 IEMS Conference*, 26, 88-97, December 2022.

[3] D. Che, D. Clark, D. Shaw, G. James. (2012). "Lessons Learned from Starting an SAE Baja Program in a Small Liberal Arts College". *ASEE*, June 2012.

[4] P. C. Wang et al., "Robotics Mentorship as a Cross-Disciplinary Platform to Foster Engineering Soft Skills", *ASEE*, June 2023.

[5] https://docs.aws.amazon.com/deepracer/latest/developerguide/deepracer-console-train-evaluate-models.html#deepracer-train-models-define-reward-function

[6] https://blog.deepracing.io/2023/03/01/join-the-aws-deepracer-league-2023/#more-1361

[7] https://today.citadel.edu/citadel-cadets-take-first-place-in-amazon-web-services-deepracer-competition-against-cadets-from-west-point-and-the-naval-academy/

Pooya Niksiar

Dr. Niksiar is assistant professor of Mechanical Engineering Department at The Citadel. He received his Ph.D. in Mechanical Engineering from Clemson University, his M.Sc. from K. N. Toosi University of Technology and his B.Sc. from Isfahan University of Technology, Iran. Prior to joining The Citadel, he was a lecturer at Clemson University. His research includes the design and development of advanced functional porous materials for bio applications. He has published several peer-reviewed journal papers in the area of his research, has received grant form NASA and Amazon and has been reviewer for several journals. Dr. Niksiar is teaching graduate and undergraduate courses in the area of thermal and fluid Sciences, aerodynamics, materials, design, measurements and numerical methods.

Blakeley Odom

Blakeley Odom is a bachelor's student in the Mechanical Engineering Department at The Citadel with an expected graduation date of May 2024. He is also part of The Citadel Master of Mechanical Engineering Program with an expected graduation date in 2025.