Assessment 1

SIT 796

Environment Introduction

An environment for a Reinforcement Learning scenario must consist of challenges that can simulate the conditions which an agent will face in a realistic setting. This case study introduces the CarRacing-v0 environment and uses the action spaces, states, and reward of the environment to argue that the environment can be used as a simulation for a realistic setting.

CarRacing-v0 environment provides a comprehensive simulated environment, which can be used to train an agent to navigate a car on a racetrack or similar roads. OpenAI Gym [1] describes the CarRacing-v0 environment as a top-down racing track, which has incoming RGB frames of size 96 x 96 pixels.

This environment is an example of continuous control task where the agent is rewarded (discussed in next section) for staying on track and finishing the race as early as possible.

State, action spaces, and reward function

As mentioned above, at any given point of time the agent can observe a state which consists of 96 x 96 pixels with two components, the racetrack, and the grass. For every episode, a random track is initialized simulating the nature of unseen roads in the real world. The objective is to drive the car on the racetrack and avoid grassy terrain. This forces the agent to look beyond the specific properties of initialized track and allows to learn the more general traits required to maneuver a car on an unseen road.

The action space is a three-length floating point tuple with components:

- Steering angle: Range [-1.0, 1.0], where -1.0 (maximum left steer possible) and +1.0 (maximum right steer possible)
- Throttle: Range [0, 1], where 0 denotes no throttle on car and 1 denotes max throttle applied.
- Brake: Range [0, 1], where 0 denotes no brakes applied and 1 denotes brakes applied with max intensity.

This action space captures the components required to drive a car in the real world. Although the action space is continuous, an agent can be trained using discrete control as well where discrete actions are mapped to continuous values which translate to a specific action, e.g. [+1.0, 0.0, 0.0] maps to turning right action. This opens the environment to a plethora of approaches that can be used to train the agent.

The reward function of the environment encourages the agent to finish the race as early as possible by giving negative (0.1) rewards per frame. The second component of the reward function encourages the agent to stay on track by giving a 1000/N positive reward per tile visited. This also means a net negative reward for straying off-track.

Summary

Through the presented action spaces and reward function, the agent develops the capabilities to maneuver on unseen terrain, handle curves, and stay on track. The continuous values of action space make the environment highly tuneable and allow the agent to learn the best values which are required for achieving a maximum reward. The configurable nature of the environment also allows adding more scenarios such as introducing obstacles and more complex racetracks for the agent to handle. This environment also allows the agent to learn complex traits such as predicting the trajectory of the moving vehicle and performing corrective actions based on it.

These properties make the CarRacing-v0 environment an ideal environment to train an agent to drive in an unseen racetrack. It is a good starting point to train an agent in capabilities required for a self-driving car, with applications in many real-world scenarios like Advanced Driver Assistance Systems (ADAS) and autonomous navigation solutions.

References

[1] Brockman, G. et al., 2016. Openai gym. *arXiv preprint arXiv:1606.01540*.