Project 1: Robot Learning Locomotion Skill for Imperfect Scenario

Problem Statement

Sensors have become an important source of data for robot applications. Unfortunately, sensor data is subjected to several sources of errors such as noise from external sources, hardware noise, inaccuracies and imprecision, various environmental effects or an adversarial perturbation to the input dimension. Numerous external and uncontrollable factors may in turn affect the quality (accuracy) of the reported sensor reading due to malfunctioning or environmental changes. If deep reinforcement learning is applied to the real-world robotics tasks, it will be important to have algorithms that are safe and robust to such input variances. For example, in robotic locomotion, the task is to walk/run. As the policy is trained on a simulated state, it is not clear what would happen if noisy input is given to the agent. Hence in this project our task is to simulate such noisy scenarios and train the agent for these situations.

Background

A generic RL problem is described as an MDP, described by state, action, reward, and dynamics of the system. While learning the policy, dynamic state-distribution causes a serious problem, as change in the state distribution will cause an oscillation in policy learning. Therefore, the policy updates are needed to be more carefully crafted. The imperceptible perturbation used for crafting malicious input for the ML model is known as adversarial perturbation. If the adversary targets the state space, imperceptible perturbations can be added to the environment directly by perturbing the sensors.

Methodology

Using model-free methods such as SAC and TD3, models will be trained on different environments. After training the model, results will be analyzed on imperfect setting by changing the environment.

Stage1:

Step 1: Setting up the simulation environment: first we need to install the gym library with mujoco setup. Mujoco is a physics based engine that builds the physics for locomotion task.

Step 2: Develop a deep reinforcement learning model: In this step different deep reinforcement learning models will be trained for different hyper-parameter settings.

Step 3: Testing and Analyzing: In this step, each model trained on different parameter setting will be evaluated for imperfect situation such as noisy state, change in friction coefficient value.

Step 4: Develop a robust model: In this step a robust model will be developed that can handle different types of noise.

Experimental Design

Dataset: To work in deep reinforcement learning we need simulation environment that can generate lots of data to work. In this project we will generate such data using OpenAI gym environment which is a python library.

Evaluation metric: Episode return (sum of reward of an episode) in nominal and noisy situation.

S/W and H/W requirements

S/W: Anaconda Python
Deep learning libraries such as sklearn, keras, tensorflow/pytorch
Python (OpenAI Gym), mujoco license key to work with mujoco environment
Matplotlib to draw comparison graphs.

H/W: A Laptop/desktop.

Project 2: Implementing Adversarial attack and Defense in DRL

Problem Statement

Deep learning models had shown degraded results on different types of attacks. Same is observed in deep reinforcement learning as DRL uses deep learning as a function approximator. In this project, we aim to implement some DRL attacks and the defense mechanisms.

Background

Prior work has shown that deep RL policies are vulnerable to small adversarial perturbations to their observations, similar to adversarial examples in image classifiers. This threat model assumes the adversary can directly modify the victim's sensory observation. Such low-level access is rarely possible. For example, an autonomous vehicle's camera image can be influenced by other drivers, but only to a limited extent.

Methodology

First we need to implement the existing adversarial attacks such as FGSM attack, enchanting attack and poison attack. After that the defense mechanism proposed in robust DRL will be used.

Step 1: Setting up the simulation environment: first we need to install the gym library in Python.

Step 2: Develop a model: In this step different deep reinforcement learning models will be developed and will be trained in one of the gym environment.

Step 3: Develop a attack: In this step, attacks such as FGSM to craft adversarial example, strategically-timed attack, enchanting and poison attack will be developed.

Step 4: Analyze the results: In this step, each trained model will be tested for different attacks and the degradation in results will be observed and analyzed.

Step 5: Develop a Defense: In this step, a defense mechanism that can tolerate such attacks will be developed and the results will be analyzed in the presence of attack.

Experimental Design

Dataset: To work in deep reinforcement learning we need simulation environment that can generate lots of data to work. In this project we will generate such data using OpenAI gym environment which is a python library.

Evaluation metric: Episode return (sum of reward of an episode) in nominal and noisy situation.

S/W and H/W requirements

S/W: Anaconda Python

Python deep learning libraries such as sklearn, keras, tensorflow/pytorch
Matplotlib to draw comparison graphs.

H/W: A Laptop/desktop with NVIDIA GPU for fast running.

Useful Links:
https://github.com/behzadanksu/rl-attack
https://www.researchgate.net/publication/318829497_Tactics_of_Adversarial_Attack_on_Deep_Reinforcement_Learning_Agents
https://paperswithcode.com/paper/learning-key-steps-to-attack-deep