ME 599: Machine Learning Control

HW 1

# Notes:

I worked on my local machine and pushed the code to github. Link is as follows: <http://github.com/vidyaaranyaM/MachineLearningControl>

## Running RL\_example notebook/python file: (Look at RL\_example.py file)

Looking at the file RL\_example.py, we see a few lines of python code and it is quite straightforward to understand what is happening in each line. Initially we import our installed “gymnasium” package and then initialize our “CartPole-v1” environment. Subsequently, we reset the environment so that everything in the environment is set to a known initial state. For our example, the initial state would consist of a cart with a pole standing vertically erect on it. The reset function returns the first observation and information about the environment. Next, we iterate for a certain number of time steps. Inside the loop, we sample a random action from the associated action space of the environment. For our example, the action space is 1 dimensional indicating the direction the cart is to be pushed - left or right. We then step the environment by taking that action which returns new observation, reward obtained, indication if the state has terminated i.e., out of bounds state, if we have reached the maximum limit for number of episodes indicated by truncated and any other required information about the environment’s internal state. At the end, we reset the environment if either it is terminated or truncated. Moreover, observation for our example includes four variables – position of cart, position of pole, velocity of cart and angular velocity of pole. After the loop is completed, we clean the processes by closing the environment.

## Trying different environments: (Look at toy\_task.py and box\_2d.py files))

Code can be found in “box\_2d.py” and “toy\_text.py” files. Tried “Bipedal Walker” environment where the goal is for a four joint walker robot to traverse as far as possible on an uneven terrain. Also tried an example from “Toy Text” environments - “Cliff Walker.” Here the goal is to reach a goal state by avoiding cliff positions. I also tried installing mujoco on mac which did not work out in the end. I installed the binaries and setup mujoco as suggested in their website but when I tried to run it integrated with gymnasium library, a message popped up suggesting to install mujoco the old way – extracting and pasting the files in mujoco/mujoco210 location. This usually worked for me earlier when I tried on an ubuntu machine but for mac, an error popped up – “RuntimeError: Could not find supported GCC executable”. I tried looking up online but it said to install older versions of gcc but that too failed as those libraries were deprecated. I tried for more than an hour and decided to work with other environments. I will give a go at it again today sometime.

## Training an Agent (Look at training\_agents folder):

I was able to setup the “Mountain Car” environment and run Reinforce training algorithm to train the agent. The goal in this environment is for the cart to gain momentum and reach a flag at the top. To the already available code online, I added functionality for saving and loading the trained models. I also added code to play the learned agent and see how it performs. Training took a while, so I did not train the agent until task completion but when looking at the behavior of the agent, it does seem to have learned to accelerate consistently. Although, looking at the reward curve would provide a metric to determine if further training is required.