For week 6, we are examining the environment of OpenAI project. OpenAI gym project contains some of classic optimal control problem. In week 6 lab, the lab uses Cart pole environment. there are 4 values in observation space: position of the cart, velocity of the cart, angular position of the end of the pole, and the angular velocity of the end of the pole. Space action contains 2 actions: 0 move cart to the left, and 1 move the cart to right. The goal is to make the pole balanced on the cart, each action results to 4 values: a new state, a reward whether the pole remains balanced or not, and Boolean value to indicate terminal state. In cart pole random policy file, the agent takes random action and counting the steps until the pole fall, then printing the number of steps that the pole remained balanced without falling, I have run the code multiple times and the number of steps is completely random and it varies from 10 to 30. The last function runs the previous process 1000 times then return the mean, minimum , and maximum number of steps that took the agent to make the pole balanced. I also ran this function multiple times and what I noticed is that the minimum number of steps remains at 8 steps, the maximum number of steps on the other hand varies from 70 to 130 steps. It is strange that the minimum number of steps is 8 when actions are taken randomly.

I went further to examine another environment from openAI. The mountain car environment contains a car between two mountains and the goal at the peak of the right mountain, the goal is to make the car reach the goal. However, the car is not strong enough to go to the right without building a momentum, to achieve that the car has to drive back and forth to build enough momentum to reach the goal. Unfortunately, due to the complete randomness, the code I have was only able to make the car reach the goal once, and the rest are the car is trying to build a momentum but without reaching the goal.

It is fascinating to see the reinforcement learning methods are used and see the animation of the agent trying taking actions and see the results.



Figure Mountain Car environment

# References

sentdex. (2019, June 5). *Q-Learning Agent Analysis - Reinforcement Learning p.3*. Retrieved from youtube: https://www.youtube.com/watch?v=CBTbifYx6a8