Jayam Sutariya

Dr. Nguyen

CS 482

5th December 2021

Project 3 Report

Introduction

For this first project, we were required to implement one of the most important Artificial Intelligence concepts we learned in class. The concept is that of Reinforcement Learning. The concept is very useful in artificial agent problems. Two such applications would be the cart pole problem and the mountain car problem. With our knowledge on Reinforcement Learning combined with some basic programming principles, the objective is to create a program that trains a model based on the problem and learns Q-values based on reward and other factors and use that model to test on scenarios and hopefully achieve success.

Explanation of approach

The program first uses argparse to parse command line arguments to select training or testing mode. The problem type (cartpole or mountain car) can also be declared through command line arguments. For the training mode, based on parameters selected in the code, a model is trained. The first problem is that of the cartpole. Not much is changed here except the reward value (gamma) and the learning rate (alpha), and also the update function for Q. Gamma is usually set higher, much closer to 1.0 and alpha is set lower, much closer to 0. Therefore, I went through multiple different values and the ones that resulted in the best results were 0.8 for gamma and 0.1 for alpha. For the update rule, I translated this equation into code and that was about it:

$$Q(a,s) \leftarrow Q(a,s) + \alpha(R(s) + \gamma * \max_{a'} Q(a',s') - Q(a,s))$$

For the next problem (mountain car), the update rule would stay the same and the alpha and gamma values would stay the same. The min and max values for the input variables also changed because it is a whole new problem. For the first problem, the bin variable was [9, 9, 9, 9] corresponding to the four input variables. This time however, since there are only two input variables, x and xdot, the bin values and the bin variable would change. I decided to go with [15, 15] as my bin variable. Since the bin values changed, the number of states would also change. Since I went with 15, the number of states would 16*16 – 1, which is 255. The number of episodes would also change. There is no need to run 50001 for this problem. Therefore, I went with 10001 episodes.

The training mode would output a .npy file containing the model information.

Results and Discussion

Using the testing mode, testing was done using the .npy file containing the model information outputted in the training mode. For the first problem, the rod had to stay upright for about 10 seconds and for the second problem, the car had to reach the flag for success. For each problem, using the aforementioned parameters, ideal results were achieved. Ran the testing mode 5 times for each problem and achieved success for every run.