**Dynamic Programming to Solve a Grid World**

**Joseph Naro**

Student at the University at Buffalo

[jfnaro@buffalo.edu](mailto:jfnaro@buffalo.edu)

**Abstract**

abahrthnartnhatrha

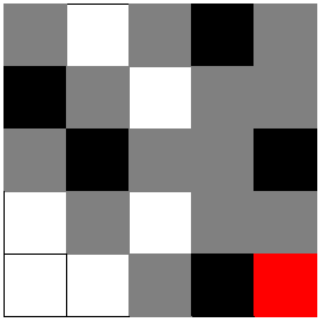
A note about running my code. I used the turtle library to render images. Uncomment the render calls if you would like to have them rendered on your computer. Doing this will open the rendering in another window and freeze the program until you close the turtle window. The code to create my environments is more robust than the directions require. This is because I wanted complete knowledge of how my environment functions, and I want to be able to reuse this code to generate new environments as I continue to learn about new methods. My environment does follow OpenAI Gym standards.

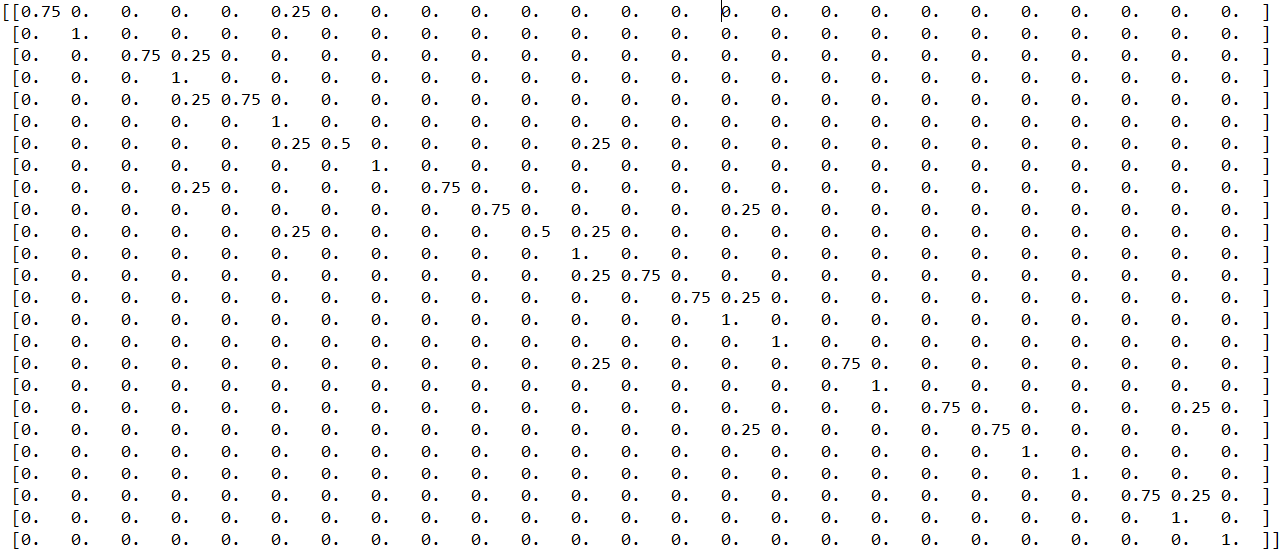
My deterministic environment is a 5 by 5 grid world with 5 holes and a final state in the bottom right. Below is a rendering I have created using the turtle library. The black spots are holes; the white spots are empty spaces; and the red spot is the final state. Each hole gives a penalty of -10, each empty space gives a penalty of -1, and the final state gives a reward of 0. There are 4 possible actions, up, right, down, and left, which will bring the agent to the state in the corresponding direction; if an action is taken into a border, the agent will remain in its current state. My agent’s objective is to get to the final state with a maximum return and avoid hole states.

A picture containing crossword puzzle, shoji, indoor, public

Description automatically generated

My stochastic environment is exactly the same as my deterministic environment, except that any action taken to a state that is not the final state and adjacent to a hole as a 25 percent chance of moving to the adjacent hole state for each adjacent hole. Below is a rendering of the stochastic environment. The colors represent the same states as those in the deterministic environment with the addition of the gray spaces, which represent the states where there is a risk of slipping into an adjacent hole, there states still give a penalty of -1.



 I have used