Neural Network Based Reinforcement Learning

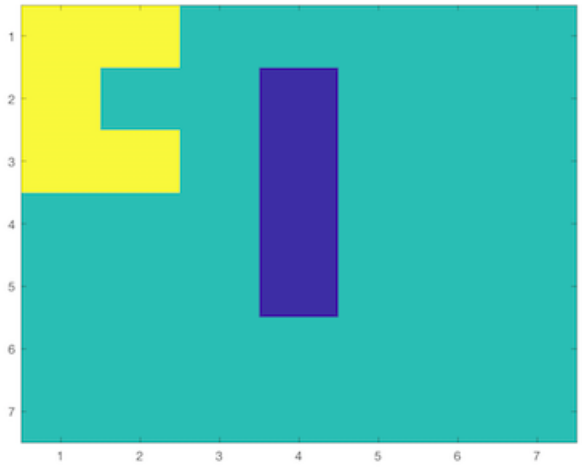
In this project, you will implement an agent that uses a neural network based reinforcement learning scheme to learn to navigate in an unstructured environment. This project will cover the following concepts from previous lectures:

- Neural Networks
- Gradient Based Learning

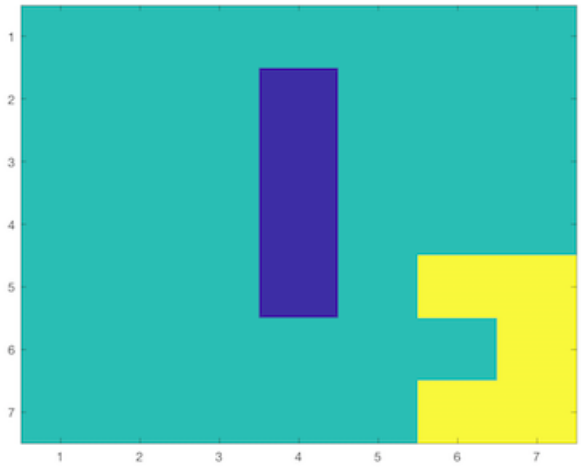
Note that before you start the project you should have completed the Week 12 Lab on neural networks. This project will allow you to learn several concepts related to:

- Reinforcement Learning
- Integrating Neural Networks into a Reinforcement Learning

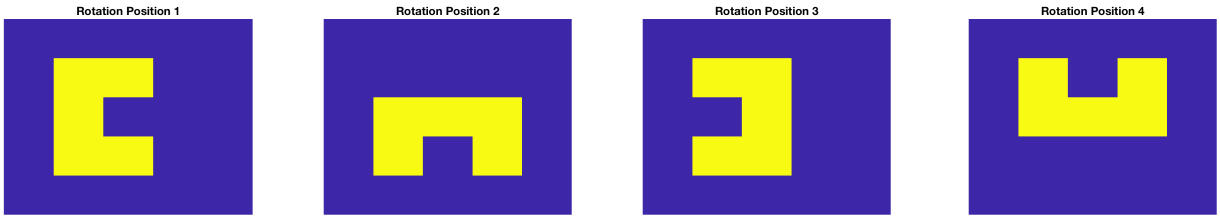
Now, imagine a robot that is deployed in an unfamiliar environment and has to reach a certain goal by navigating in this environment. The entire environment where the robot navigates can be viewed as a 2D grid illustrated below:



where the yellow structure indicates a robot itself, and a blue color indicates the wall that the robot cannot pass through. Our goal is to achieve the following state:



At every step, the robot can move {up, down, right, left}. Furthermore, the robot can also rotate to one of the following four positions displayed below. The valid transitions include rotating from rotation position 1 to positions 2 and 4, rotating from rotation position 2 to positions 1 and 3, rotating from rotation position 3 to positions 2 and 4, and rotating from rotation position 4 to positions 3 and 1.



Then, the robot needs to dynamically figure out what is the best way to reach its goal state by continuously exploring the environment and learning about it. We will achieve this goal using a neural network based reinforcement learning scheme.

Intuitively, the main principle of reinforcement learning is as follows. The robot is allowed to navigate across the grid and explore the environment. However, every time the robot hits the wall it receives a negative feedback from the environment. Furthermore, every time the robot reaches the goal state, it receives a reward. Then, our goal is to train a neural network that would examine a current robot's state, and decide what the next action should be. Ideally, we want to select actions, which would have a high probability of leading to the reward and low probability of leading to the negative feedback. In the context of our problem, we want the robot to learn to navigate the environment (select actions) such that it could reach the goal state (reward) without hitting the walls (negative feedback).

In case you want to work on this problem outside of the EdX environment, we provide the required files in the following zip file: Project4.zip. Note that we use slightly different neural network functions than the ones implemented in Lab12. Therefore, we provide those functions in the zip file as well.

Note that this lab builds on top of each function that you will be implementing. Therefore, if you are working outside of the EdX environment, you need to complete each part of this lab before proceeding to another one. If you are working inside the EdX environment, it is highly recommended but not necessary to complete one part before starting another one. That is, in the end, even if your code in the earlier part is incorrect, you can still get the full credit for the later parts if they are implemented correctly.