# CST8509 Lab2 CliffWalking

## Overview

Q-learning is a popular Temporal Difference method for Reinforcement Learning. In this lab, you will follow existing examples of a Q-learning implementation, and then modify one of those implementations to solve the CliffWalking grid world as described in the Sutton textbook on Page 132.

When you have completed this lab, you will know how to

* Explain how Temporal Difference learning (specifically Q-learning) works
* Apply Q-Learning to various grid world examples.

## Instructions

Complete Hybrid Activity 2: this is the starting tutorial for this lab exercise. You don’t need to create an account – close that window if it appears. The URL is <https://medium.com/data-science/math-of-q-learning-python-code-5dcbdc49b6f6>

The beginning of the tutorial will be review for you and you should be able to skim over it. The rest of the tutorial is new material that explains the Bellman equation, which we will revisit in class in the coming weeks. Overall, this tutorial forms the basis for our work with Q-learning grid world examples.

Your task is to

* After Hybrid Activity 2, download the provided source code files, and make these changes:
  + Rename medium\_qlearning\_rl.py to **<your\_algonquin\_id>\_lab2\_qlearning\_rl.py**
  + Rename medium\_qlearning\_env.py to **<your\_algonquin\_id>\_lab2\_env.py**
  + Change the import at the top of **<your\_algonquin\_id>\_lab2\_qlearning\_rl.py** to reflect the name change
* Read the python code. You should have an understanding of how the code works before you run it.
* Run the example code, and notice that the agent gets better and better at traversing the simple grid world.
* Make changes to the example code solve a different grid world problem, the Cliff Walking problem on Page 132 of the Sutton Textbook, according to these instructions:
  + Do not use an existing solution of the Cliff Walking world – create a new solution by changing the Hybrid 2 example grid world according to the points below.
  + Copy **<your\_algonquin\_id>\_lab2\_env.py** to **<your\_algonquin\_id>\_lab2\_cliff\_env.py**
  + As you make the following changes to **<your\_algonquin\_id>\_lab2\_cliff\_env.py**, add a comment in your own words to explain any changes you make.
  + Under **#Hyperparameters** add **alpha=1** step-size hyperparameter to be discussed during your demonstration
  + Change the **Env** class initializer to represent the shape of the Cliff Walking world, and introduce a “cliff” attribute (True/False) to indicate whether the agent fell over the cliff while performing the last action
  + Change the **step** method according to the Cliff Walking world. If that step results in falling off the cliff, the **cliff** attribute should be set to **True**; otherwise, the **cliff** attribute should be **False**. Make sure the resulting state is correct if the agent falls off the cliff.
  + Change the **reward** calculation to match the Cliff Walking world.
  + Change the **render** method to indicate the cliff part of the grid (the ten grey squares) as X’s
  + Change the “epoch” term to our preferred term for this context: “episode”
  + Print the Return (total accumulated reward) under the number of steps when the goal is reached.

## Submission

* Submit a zip file containing (a folder of) the all of the python source code files.

## Demonstration

* Show your lab instructor your running Cliff World solution.
* Be prepared to discuss briefly some aspects of your changes you made to the original code.