Q-LEARNING

Kevin Chen and Zack Khan

University of Maryland

CMSC389F: Reinforcement Learning, Spring 2018

MIDTERM GRADES

Released this weekend on CS Grade Server!

TEAM ASSIGNMENTS

Team A: Johann Miller, Ibrahim Jirdeh, Dhruv Mehta, Atharva Bhat

Team B: Tianrui Guan, Yexin Wu, Nguyen Minh

Team C: Christopher Yue, Vishal Hundal, Acheev Bhagat

Team D: Alex Bloch, Carl Brando

Team E: Wei-Hsuan Lin, Kaung Thant

Team F: David Kaplan, Graham Buck

Team G: Timothy Zhou, Catherine Lau, and Stephan Loh.

Team H: Jaemoon Park, Justin Becker

Team I: Cindy Guijosa, Mingda Guo, Joshua Karper, Cameron Long

Team J: Baram Sosis, Eric Wallace, Yifei Yu

PROJECT

We will have several *coding lectures* where we code up RL algorithms in-class
(This week is Q-Learning)

We will have a few research-oriented lectures including:

1. Useful plots to generate for RL
2. Quick overview of landmark RL papers
3. Quick overview of DQN

The Due Date for the project is **May 11th**

PROJECT

A 4-6 page research paper on an RL topic of your choice.

Introduction (½ page): Summary of the paper

Background (1 page): Necessary information to understand your paper (i.e lecture material)

Results (1-2 pages): Tables and plots about your results

Discussion (1 page): Discuss your results

Conclusion (½ page): Conclude everything

PROJECT

A 4-6 page research paper on an RL topic of your choice.

Introduction (½ page): Summary of the paper

Background (1 page): Necessary information to understand your paper (i.e lecture material)

Results (1-2 pages): Tables and plots about your results

Discussion (1 page): Discuss your results

Conclusion (½ page): Conclude everything

PROJECT

A 4-6 page research paper on an RL topic of your choice.

Introduction (½ page): Summary of the paper

Background (1 page): Necessary information to understand your paper (i.e lecture material)

Results (1-2 pages): Tables and plots about your results

Discussion (1 page): Discuss your results

Conclusion (½ page): Conclude everything

PROJECT

A 4-6 page research paper on an RL topic of your choice.

Introduction (½ page): Summary of the paper

Background (1 page): Necessary information to understand your paper (i.e lecture material)

Results (1-2 pages): Tables and plots about your results

Discussion (1 page): Discuss your results

Conclusion (½ page): Conclude everything

PROJECT

A 4-6 page research paper on an RL topic of your choice.

Introduction (½ page): Summary of the paper

Background (1 page): Necessary information to understand your paper (i.e lecture material)

Results (1-2 pages): Tables and plots about your results

Discussion (1 page): Discuss your results

Conclusion (½ page): Conclude everything

EXAMPLE PROJECTS

A 4-6 page research paper on an RL topic of your choice.

Introduction (½ page): Summary of the paper

Background (1 page): Necessary information to understand your paper (i.e lecture material)

Results (1-2 pages): Tables and plots about your results

Discussion (1 page): Discuss your results

Conclusion (½ page): Conclude everything

TEAM ASSIGNMENTS

Take 10 minutes to meet with your teams and discuss potential project ideas.

Check out the different environments you can use here: https://gym.openai.com/envs/#classic_control

Feel free to use environments from any of the categories except Robotics and MuJuCo (which require dedicated GPUs)

Think about what kind of questions you want to explore and what algorithms we've learned that you want to implement

OFFICE HOURS

When: Friday 3-4pm

Where: ESJ 1232 (Today Only)

Check Piazza for updated location each week!

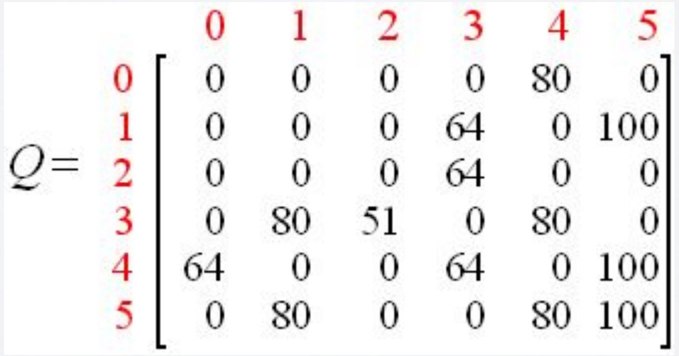
Q-LEARNING

INTUITION

Similar to TD Learning, we will also use the actual cumulative reward we get from successor states to update the value of previous states.

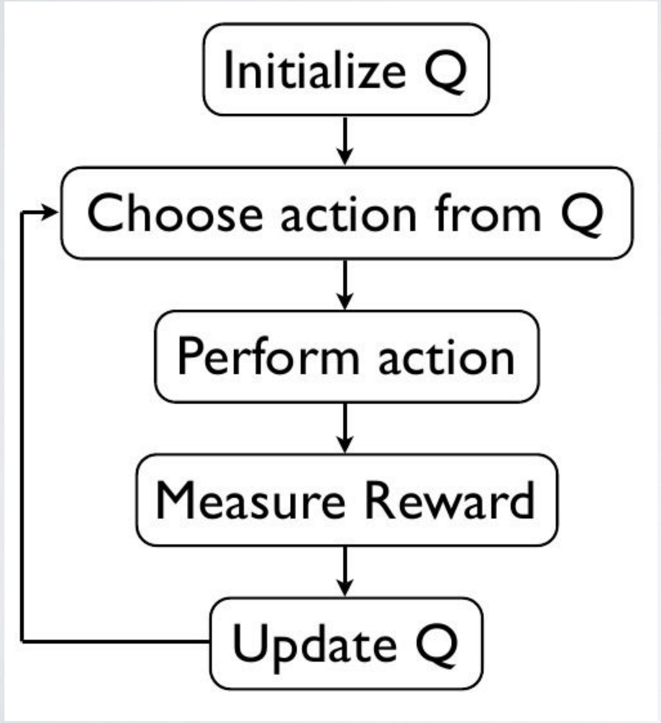
This time, we will do this with action values: Q(s,a)

We keep track of a Q table like this:



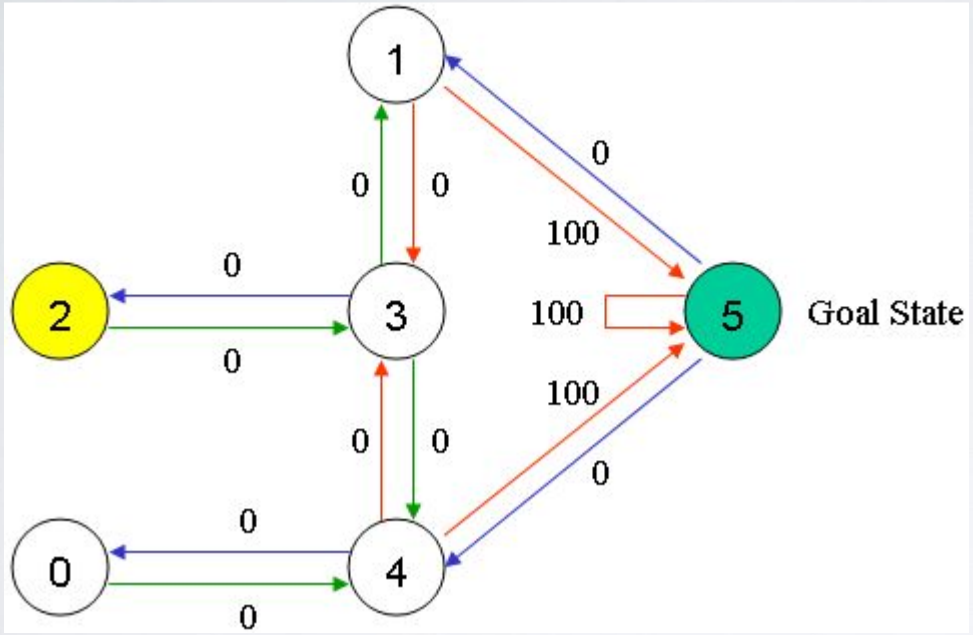
$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 80 & 0 \\ 0 & 0 & 0 & 64 & 0 & 100 \\ 0 & 0 & 0 & 64 & 0 & 0 \\ 0 & 80 & 51 & 0 & 80 & 0 \\ 64 & 0 & 0 & 64 & 0 & 100 \\ 0 & 80 & 0 & 0 & 80 & 100 \end{bmatrix} \end{array}$$

Q-LEARNING ALGORITHM



1. Initialize the Values table Q(s, a)
2. Choose current state s (can be chosen randomly)
3. Choose an action a for the next state greedily based on Q Values in table
4. Take the action, and observe the reward r as well as the new state s’
5. Update the Value for the state using the observed reward and the maximum reward possible for the next state: **Q(state, action) = R(state, action) + Gamma * Max[Q(s’, a)] for all actions a**
6. Set the current state to the new state, and repeat the process until a terminal state is reached.

Example



Gamma = 0.8

$$
R = \begin{array}{c|cccccc}
\text{State} \backslash \text{Action} & 0 & 1 & 2 & 3 & 4 & 5 \\
\hline
0 & -1 & -1 & -1 & -1 & 0 & -1 \\
1 & -1 & -1 & -1 & 0 & -1 & 100 \\
2 & -1 & -1 & -1 & 0 & -1 & -1 \\
3 & -1 & 0 & 0 & -1 & 0 & -1 \\
4 & 0 & -1 & -1 & 0 & -1 & 100 \\
5 & -1 & 0 & -1 & -1 & 0 & 100
\end{array}
$$

$$
Q = \begin{array}{c|cccccc}
 & 0 & 1 & 2 & 3 & 4 & 5 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Example: Episode 1

First episode:

Start at state 1.
Evaluate all possible state action pairs from state 1: can either go to state 3 or 5.

Since both Q(1,3) and Q(1,5) are 0 in my table, I can pick either one. Let’s choose 5 randomly, so we are going to update Q(1,5).

Recall: **Q(state, action) = R(state, action) + Gamma * Max[Q(s’, a)] for all actions a.**

What actions can we take from 5? We can go to 1, 4 or stay at 5, which all have a Q value of 0.

Q(1, 5) = R(1, 5) + 0.8 * Max[Q(5, 1), Q(5, 4), Q(5, 5)]
= 100 + 0.8 * 0 = 100
Thus, Q(1,5) gets updated to 100.
We stop at 5 since 5 is the goal/terminal state

$$\text{Action}$$

$$R = \begin{array}{c|cccccc} \text{State} & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 0 & -1 & -1 & -1 & -1 & 0 & -1 \\ 1 & -1 & -1 & -1 & 0 & -1 & 100 \\ 2 & -1 & -1 & -1 & 0 & -1 & -1 \\ 3 & -1 & 0 & 0 & -1 & 0 & -1 \\ 4 & 0 & -1 & -1 & 0 & -1 & 100 \\ 5 & -1 & 0 & -1 & -1 & 0 & 100 \end{array}$$

$$Q = \begin{array}{c|cccccc} & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Example: Episode 2

Second Episode

Start at state 3.
Evaluate all possible state action pairs from state 3: can go to state 1, 2, or 4.

Since Q(3,1), Q(3,2) and Q(3,4) all have values of 0, lets randomly pick Q(3,1) to update.

Recall: **Q(state, action) = R(state, action) + Gamma * Max[Q(s', a)] for all actions a.**

What actions can we take from successor state 1? We can go to 3 or 5. We calculated Q(1,5) to be 100 in the last episode.

Q(3,1) = R(3, 1) + 0.8 * Max[Q(1,3), Q(1,5)] = 0 + .8 * 100 = 80
Thus, Q(3,1) gets updated to 80.

$$
\begin{array}{cc}
 & \textbf{Action} \\
\textbf{State} & \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array}
\end{array}
$$

$$
R = \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{bmatrix}
-1 & -1 & -1 & -1 & 0 & -1 \\
-1 & -1 & -1 & 0 & -1 & 100 \\
-1 & -1 & -1 & 0 & -1 & -1 \\
-1 & 0 & 0 & -1 & 0 & -1 \\
0 & -1 & -1 & 0 & -1 & 100 \\
-1 & 0 & -1 & -1 & 0 & 100
\end{bmatrix}
$$

$$
\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array}
$$

$$
Q = \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 100 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 80 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

Example: Episode 2 (Continued)

Second Episode

Thus, Q(3,1) gets updated to 80.

Since we are currently at state 1, and 1 is not a terminal state, we must keep going.

At state 1, we can go to either 3 or 5. We pick 5.

Q(1, 5) = R(1, 5) + 0.8 * Max[Q(5, 1), Q(5, 4), Q(5, 5)]
= 100 + 0.8 * 0 = 100

Q(1,5) was already 100, so our Q table does not change.

Finally, since 5 is a terminal state, we can end our episode here.

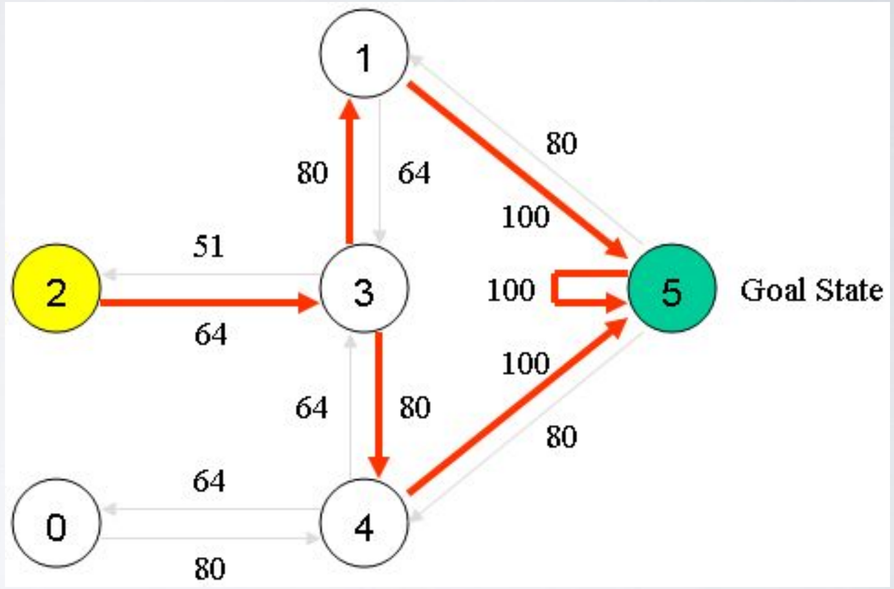
$$
\begin{array}{cc|cccccc}
 & & & & \textbf{Action} & & & \\
 & \textbf{State} & 0 & 1 & 2 & 3 & 4 & 5 \\
\hline
 & 0 & -1 & -1 & -1 & -1 & 0 & -1 \\
 & 1 & -1 & -1 & -1 & 0 & -1 & 100 \\
R= & 2 & -1 & -1 & -1 & 0 & -1 & -1 \\
 & 3 & -1 & 0 & 0 & -1 & 0 & -1 \\
 & 4 & 0 & -1 & -1 & 0 & -1 & 100 \\
 & 5 & -1 & 0 & -1 & -1 & 0 & 100
\end{array}
$$

$$
\begin{array}{cc|cccccc}
 & & 0 & 1 & 2 & 3 & 4 & 5 \\
\hline
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 1 & 0 & 0 & 0 & 0 & 0 & 100 \\
Q= & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 3 & 0 & 80 & 0 & 0 & 0 & 0 \\
 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 5 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Final Outcome

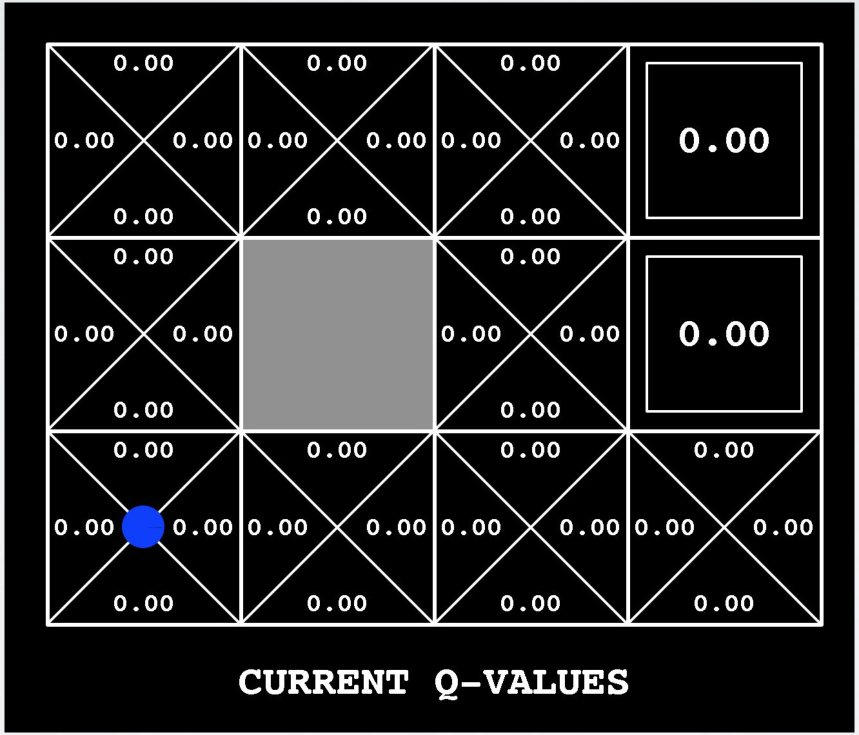
$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 80 & 0 \\ 0 & 0 & 0 & 64 & 0 & 100 \\ 0 & 0 & 0 & 64 & 0 & 0 \\ 0 & 80 & 51 & 0 & 80 & 0 \\ 64 & 0 & 0 & 64 & 0 & 100 \\ 0 & 80 & 0 & 0 & 80 & 100 \end{array}\right] \end{array}$$

(Scaled by 500)



Follow actions with highest Q value!

More examples



Exploration vs Exploitation

If we are always taking the action for the maximum Q value, how will we encourage our agent to **explore** unvisited states?

Exploration: evaluating unvisited or infrequently visited states
Exploitation: choosing best actions based off what you already know (Ex: Q table)

Potential Solution: Occasionally choose a random action instead of greedily picking the optimal action based of Q(s,a)

Coding Q-Learning