



Deep RL Course documentation
A Q-Learning example ▾



A Q-Learning example

To better understand Q-Learning, let's take a simple example:



- You're a mouse in this tiny maze. You always **start at the same starting point**.
- The goal is to **eat the big pile of cheese at the bottom right-hand corner** and avoid the poison. After all, who doesn't like cheese?
- The episode ends if we eat the poison, **eat the big pile of cheese**, or if we take more than five steps.
- The learning rate is 0.1
- The discount rate (γ) is 0.99

Example



- You always start at the **same starting point**.
- The goal: eat the **big pile of cheese** (at the bottom right-hand corner) and **avoid the poison**.
- The episode ends if we eat the poison, eat the big pile of cheese or if we spent more than 5 steps.
- Learning rate = 0.1
- Gamma = 0.99

The reward function goes like this:

- **+0**: Going to a state with no cheese in it.
- **+1**: Going to a state with a small cheese in it.
- **+10**: Going to the state with the big pile of cheese.
- **-10**: Going to the state with the poison and thus dying.
- **+0** If we take more than five steps.

Example

- The reward function:
 - 0: Going to a state with no cheese in it.
 - +1: Going to a state with a small cheese in it.
 - +10: Going to the state with the big pile of cheese.
 - -10: Going to the state with the poison and thus die.













To train our agent to have an optimal policy (so a policy that goes right, right, down), we will use the Q-Learning algorithm.

Step 1: Initialize the Q-table

Example, Step 1

Initialize Q arbitrarily (e.g., $Q(s, a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$, and $Q(\text{terminal-state}, \cdot) = 0$)

				
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0

We initialize the Q-Table

So, for now, our Q-table is useless; we need to train our Q-function using the Q-Learning algorithm.

Let's do it for 2 training timesteps:

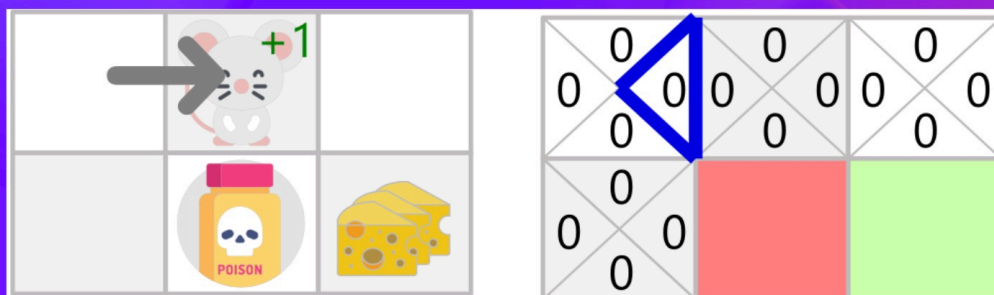
Training timestep 1:

Step 2: Choose an action using the Epsilon Greedy Strategy

Because epsilon is big ($= 1.0$), I take a random action. In this case, I go right.

Example, Step 2

Choose action A_t using policy derived from Q (e.g., ϵ -greedy)



We took a random action (exploration)

Step 3: Perform action A_t , get R_{t+1} and S_{t+1}

By going right, I get a small cheese, so $R_{t+1} = 1$ and I'm in a new state.

Example, Step 3

Take action A_t and observe R_{t+1}, S_{t+1}



Step 4: Update $Q(S_t, A_t)$

We can now update $Q(S_t, A_t)$ using our formula.

Example, Step 4

$$\underline{Q(S_t, A_t)} \leftarrow \underline{Q(S_t, A_t)} + \alpha [\underline{R_{t+1}} + \gamma \underline{\max_a Q(S_{t+1}, a)} - \underline{Q(S_t, A_t)}]$$

New
Q-value
estimation

Former
Q-value
estimation

Learning
Rate

Immediate
Reward

Discounted Estimate
optimal Q-value
of next state

Former
Q-value
estimation

TD Target

TD Error







Update our Q-value estimation

Example, Step 4

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t))$$

$$Q(\text{Initial state, Right}) = 0 + 0.1 * [1 + 0.99 * 0 - 0]$$

$$Q(\text{Initial state, Right}) = 0.1$$

	←	→	↑	↓
	0	0.1	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0

Training timestep 2:

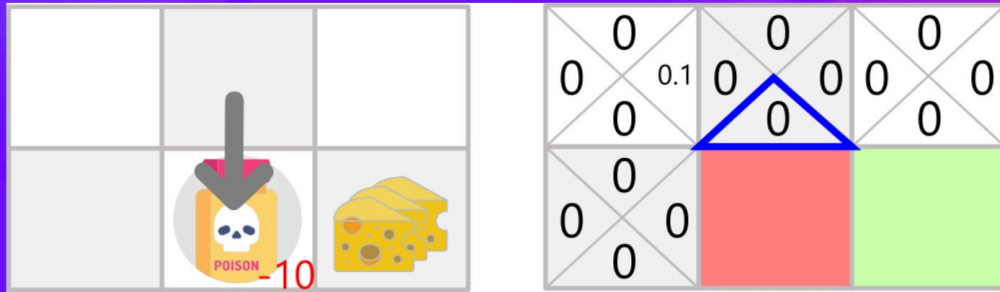
Step 2: Choose an action using the Epsilon Greedy Strategy

I take a random action again, since epsilon=0.99 is big. (Notice we decay epsilon a little bit because, as the training progress, we want less and less exploration).

I took the action 'down'. This is not a good action since it leads me to the poison.

Example, Step 2

Choose action A_t using policy derived from Q (e.g., ϵ -greedy)



We took a random action (exploration)

Step 3: Perform action A_t , get R_{t+1} and S_{t+1}

Because I ate poison, I get $R_{t+1} = -10$, and I die.

Example, Step 3

Take action A_t and observe R_{t+1}, S_{t+1}









Step 4: Update Q(St, At)

Example, Step 4

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t))$$

$$Q(\text{State 2, Down}) = 0 + 0.1 * [-10 + 0.99 * 0 - 0]$$

$$Q(\text{State 2, Down}) = -1$$

	←	→	↑	↓
	0	0.1	0	0
	0	0	0	-1
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0

Because we're dead, we start a new episode. But what we see here is that, **with two explorations steps, my agent became smarter.**

As we continue exploring and exploiting the environment and updating Q-values using the TD target, the Q-table will give us a better and better approximation. At the end of the training, we'll get an estimate of the optimal Q-function.