Module V: Numerical on Q- learning and SARSA Algorithm

Dimple Bohra

SARSA Algorithm

Consider the following Q[S,A] table:

	Action 1	Action 2	Action 3
State 1	1.5	4	1
State 2	2.5	3	1.5
State 3	2	4	3

Assume that α=0.1, and γ=0.5.
Update the Q table with the following (S, A, R, S', A') experiences using SARSA..

1.⟨1, 1, 5, 2, 1⟩

solution -> Q(1,1) = Q[s,a] + α(r + γQ[s',a'] - Q[s,a])

= 1.5 + 0.1(5 + 0.5(2.5) - 1.5)

= 1.975

At each step, state which value of the table gets updated and draw the final updated Q[S,A] table.

Q- learning Algorithm

Consider the following Q[S,A] table:

	Action 1	Action 2	Action 3
State 1	1.5	4	1
State 2	2.5	3	1.5
State 3	2	4	3

Assume that α=0.1, and γ=0.5.
Update the Q table with the following (S, A, R, S') experiences using Q learning..

- $\langle 1, 1, 5, 2 \rangle$

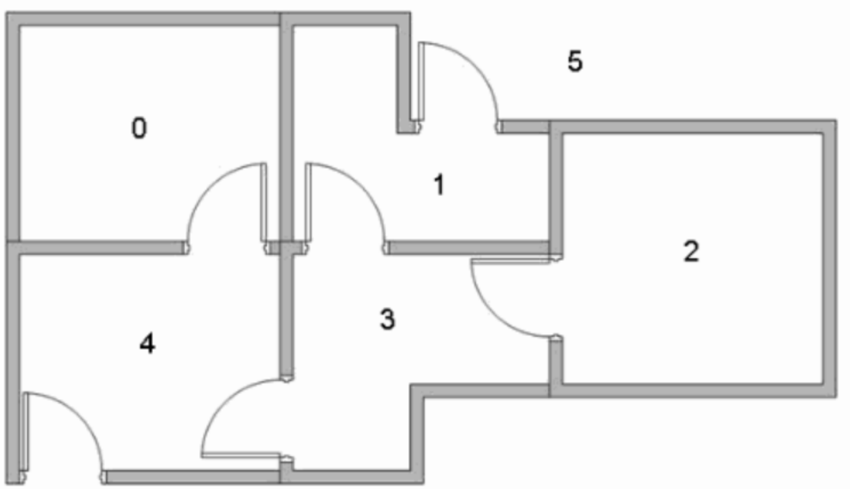
solution -> Q(1,1) = Q[s,a] + α(r + γmax$_{a'}$ Q[s',a'] - Q[s,a])

$$= 1.5 + 0.1(5 + 0.5(3) - 1.5)$$

$$= 2$$

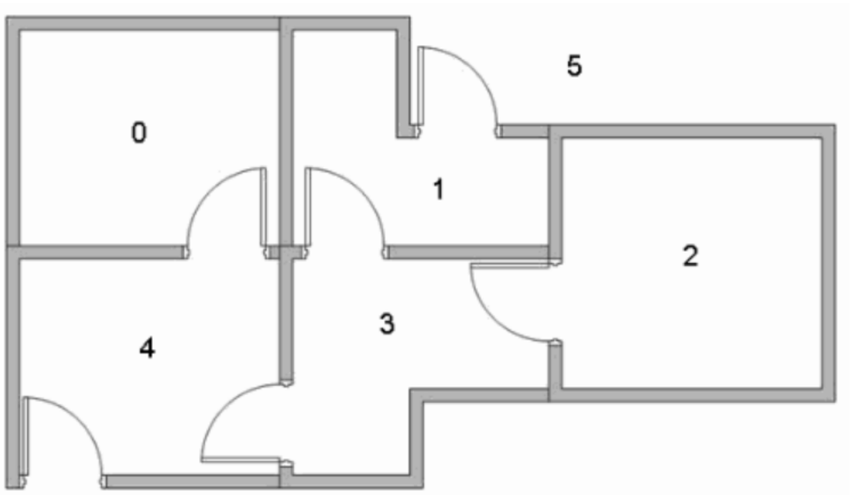
Q- Learning Example

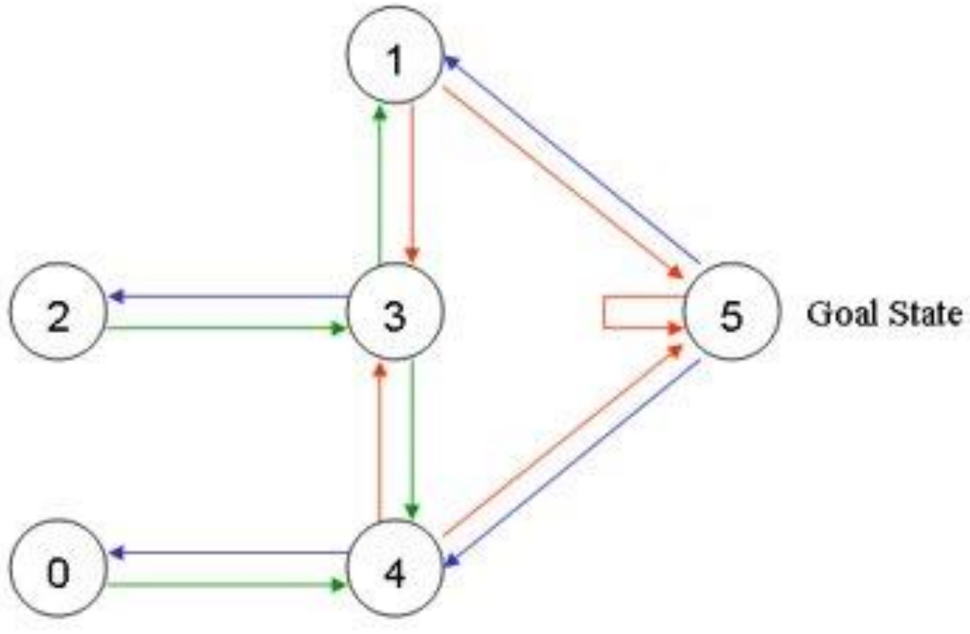
- Suppose we have 5 rooms in a building connected by doors as shown in the figure below. We'll number each room 0 through 4. The outside of the building can be thought of as one big room (5). Notice that doors 1 and 4 lead into the building from room 5 (outside).

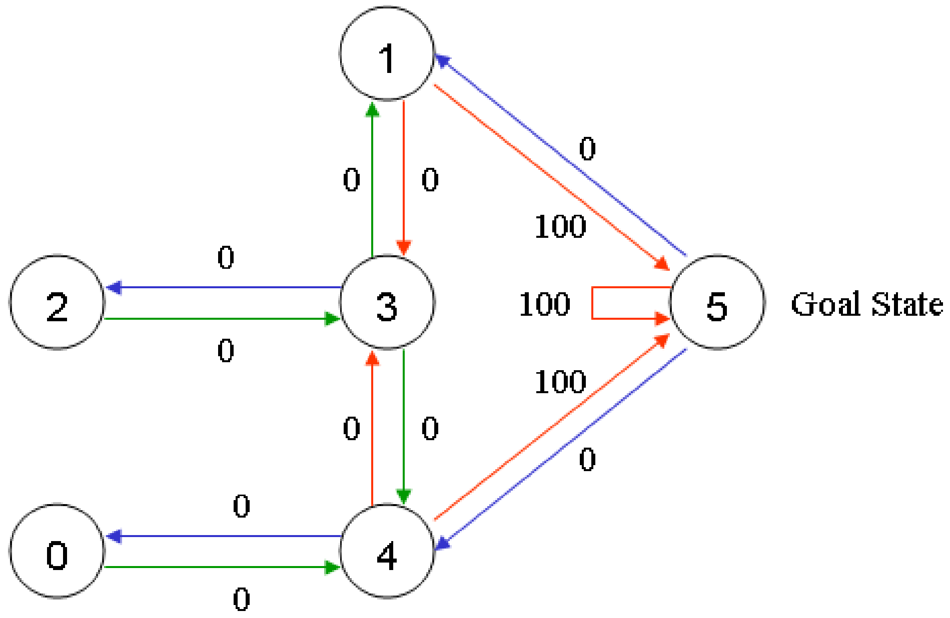


For this example, we'd like to put an agent in any room, and from that room, go outside the building (this will be our target room). Find Q values which will suggest the actions agent can take to come to target room from any room in the building.

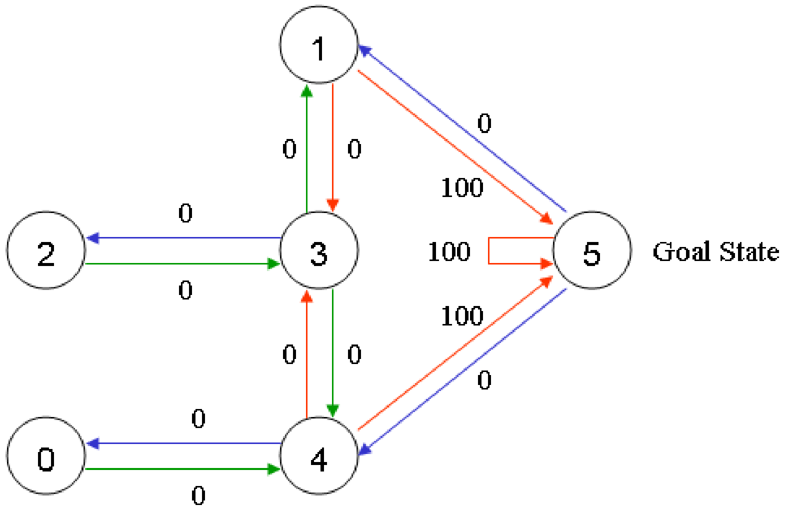
Solution







Reward to door directly connected to goal room : 100
Reward to doors not directly connected to goal room : 0

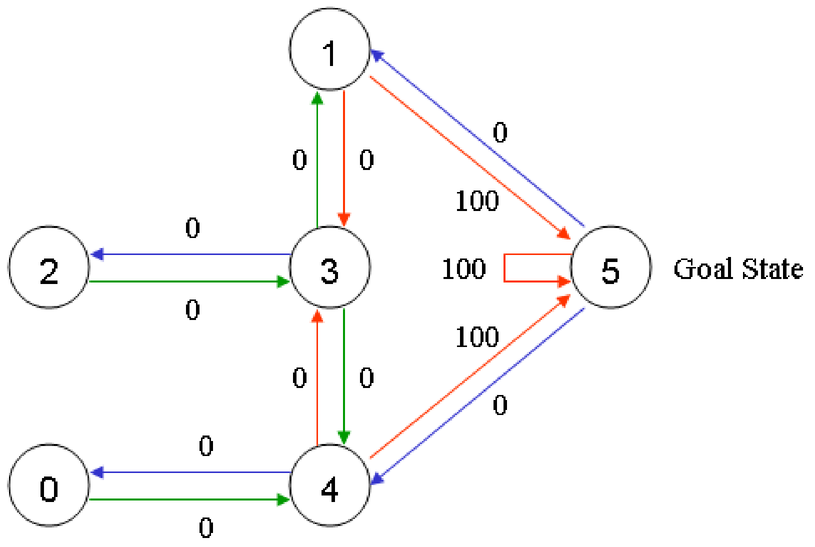


$$R=\begin{array}{c} \\ \text{State} \\ 0\\1\\2\\3\\4\\5\end{array}\begin{array}{c}\text{Action}\\ \begin{array}{cccccc}0&1&2&3&4&5\end{array}\\ \begin{bmatrix}-1&-1&-1&-1&0&-1\\-1&-1&-1&0&-1&100\\-1&-1&-1&0&-1&-1\\-1&0&0&-1&0&-1\\0&-1&-1&0&-1&100\\-1&0&-1&-1&0&100\end{bmatrix}\end{array}$$

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

Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]

Gamma = 0.8
And alpha =1



Action

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

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

Lets start with state 1:

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

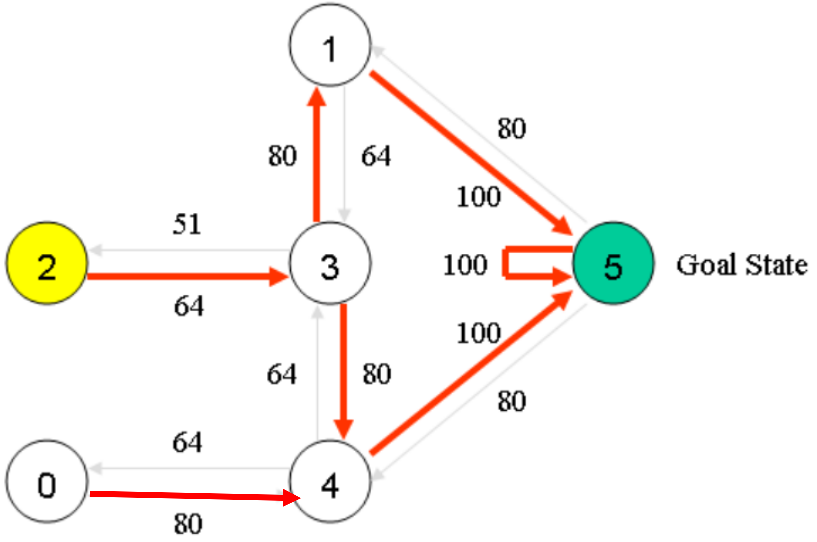
Q(3,1) = R(3,1) +0.8* max[Q(1,3),Q(1,5)]
= 0+0.8*max(0,100)= 0+0.8*100= 80

Q(2,3)= R(2,3)+0.8*max[Q(3,1),Q(3,2),Q(3,4)]
= 0+0.8*80=64

And so on... same for all remaining 10 links

Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]

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



Answer: Q table or graph gives agent which action should be taken to move towards goal by checking the maximum Q value for the current state