

The bellman equation

The Bellman Equation

Artificial Intelligence A-Z

© SuperDataScience

The Bellman Equation

Concepts:



Artificial Intelligence A-Z

© SuperDataScience

The Bellman Equation

Concepts:

- s – State

The Bellman Equation

Concepts:

- s – State
- a – Action

The Bellman Equation

Concepts:

- s – State
- a – Action
- R – Reward

Artificial Intelligence A-Z

© SuperDataScience

reward is given when agent goes to a specific state

The Bellman Equation

Concepts:

- s – State
- a – Action
- R – Reward
- γ – Discount

Artificial Intelligence A-Z

© SuperDataScience

gamma

The Bellman Equation



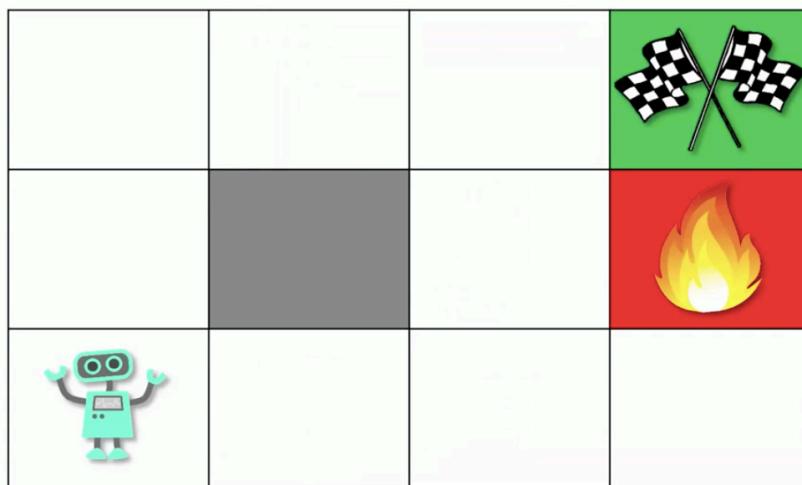
Richard Ernest Bellman

Artificial Intelligence A-Z

© SuperDataScience

Person behind reinforcement learning

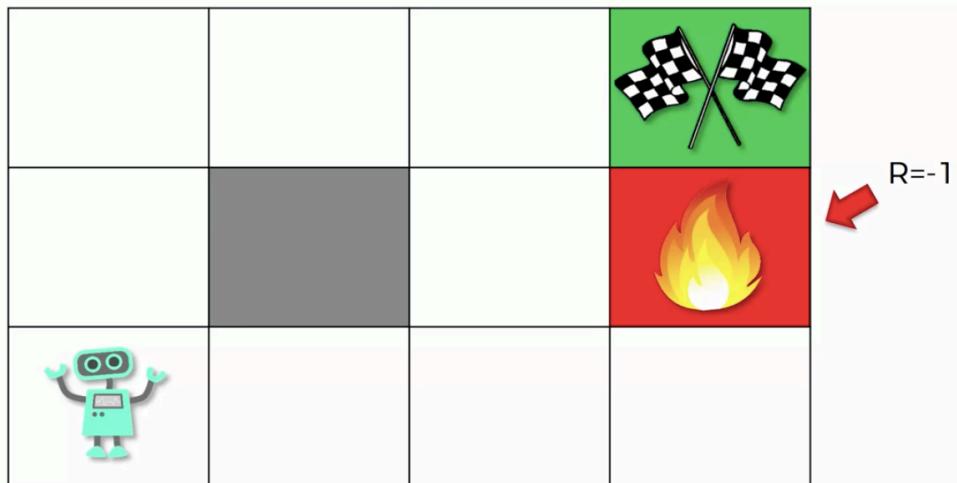
The Bellman Equation



Artificial Intelligence A-Z

© SuperDataScience

The Bellman Equation

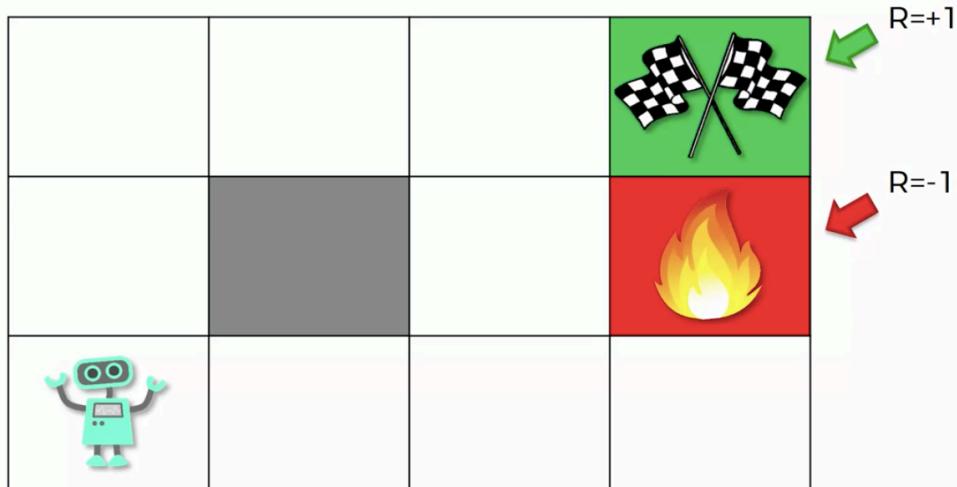


Artificial Intelligence A-Z

© SuperDataScience

-1 is punishment which we say -1 reward

The Bellman Equation

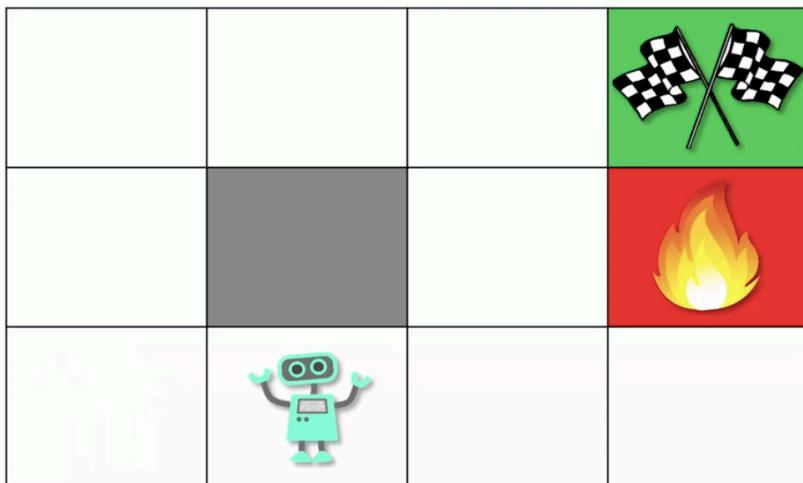


Artificial Intelligence A-Z

© SuperDataScience

plus one reward or +1 reward

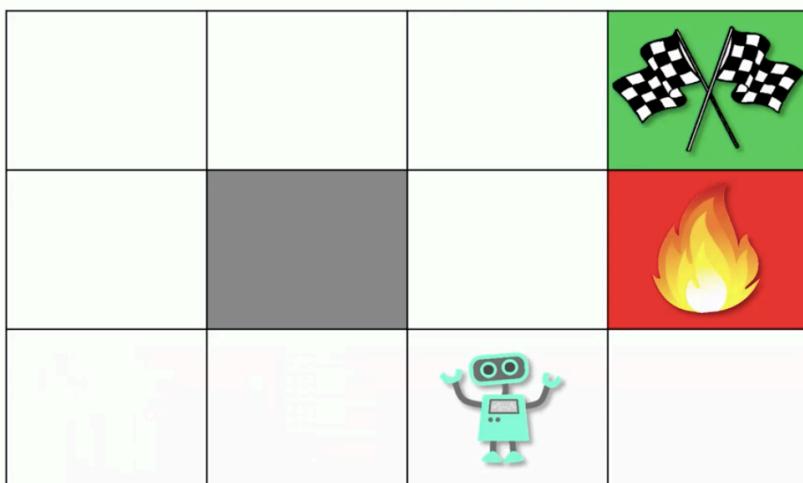
The Bellman Equation



Artificial Intelligence A-Z

© SuperDataScience

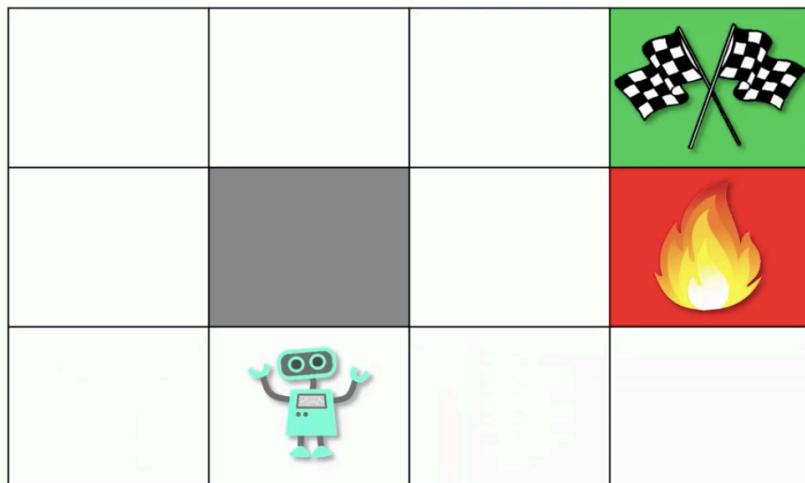
The Bellman Equation



Artificial Intelligence A-Z

© SuperDataScience

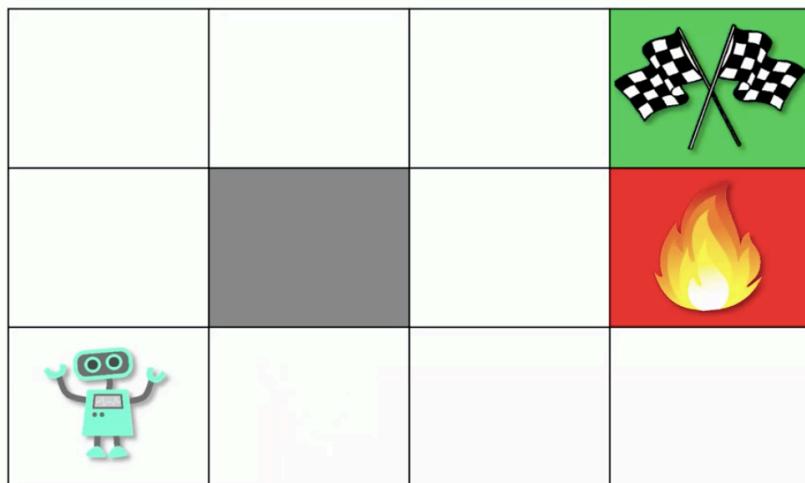
The Bellman Equation



Artificial Intelligence A-Z

© SuperDataScience

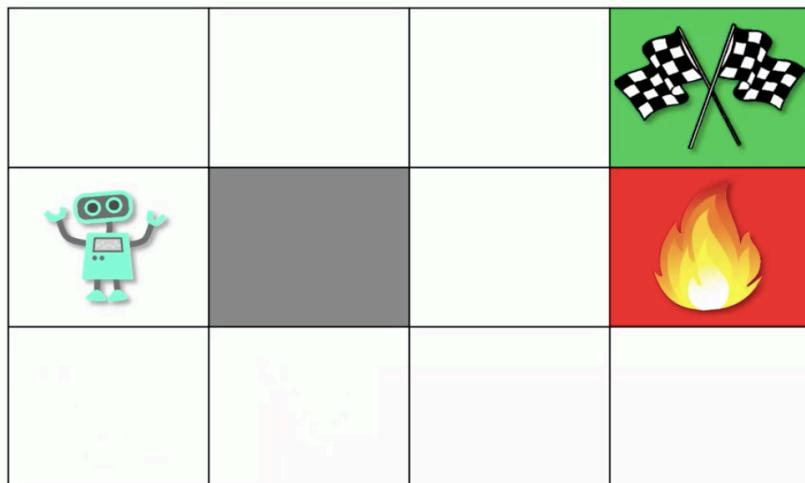
The Bellman Equation



Artificial Intelligence A-Z

© SuperDataScience

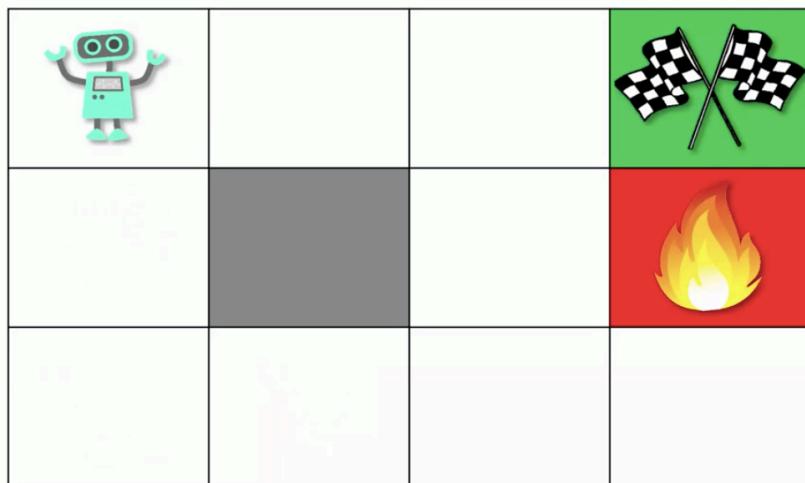
The Bellman Equation



Artificial Intelligence A-Z

© SuperDataScience

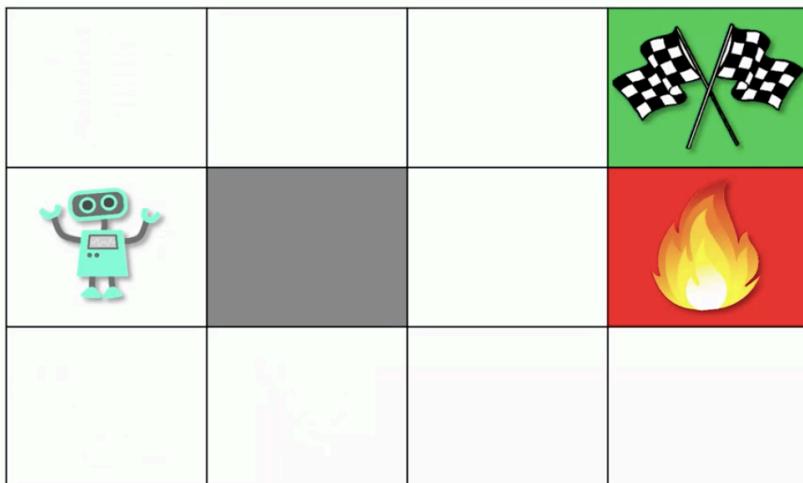
The Bellman Equation



Artificial Intelligence A-Z

© SuperDataScience

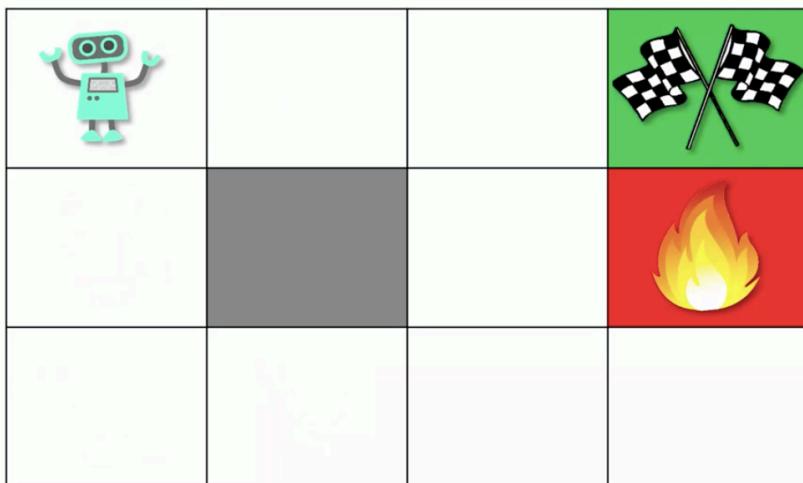
The Bellman Equation



Artificial Intelligence A-Z

© SuperDataScience

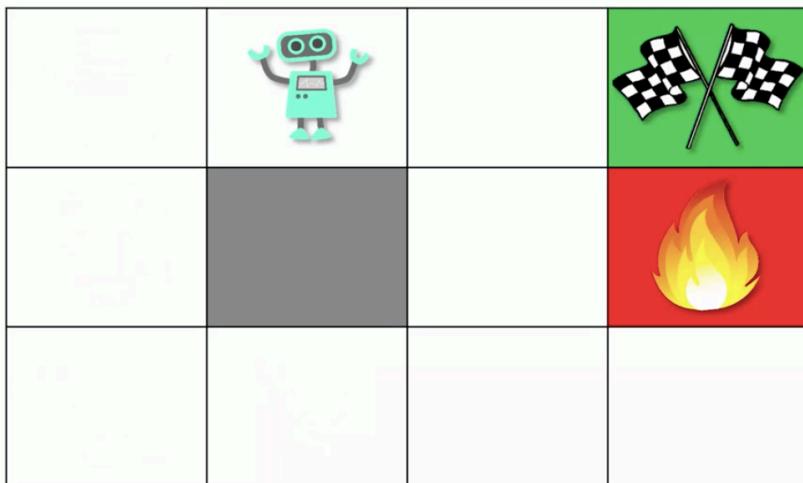
The Bellman Equation



Artificial Intelligence A-Z

© SuperDataScience

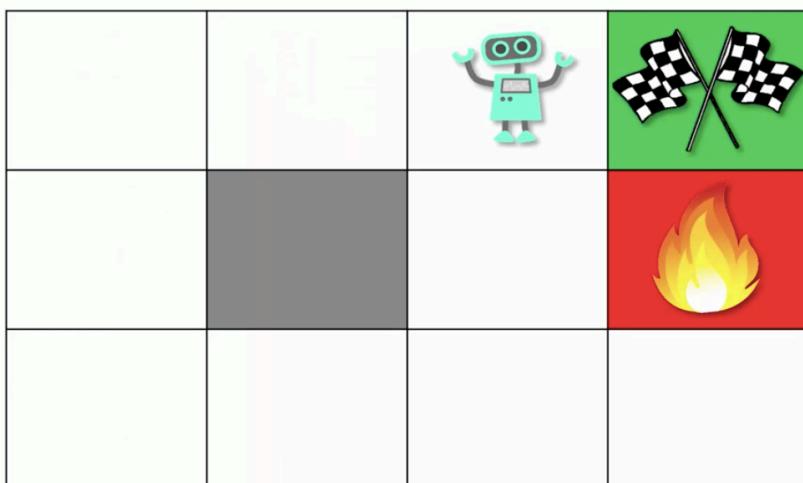
The Bellman Equation



Artificial Intelligence A-Z

© SuperDataScience

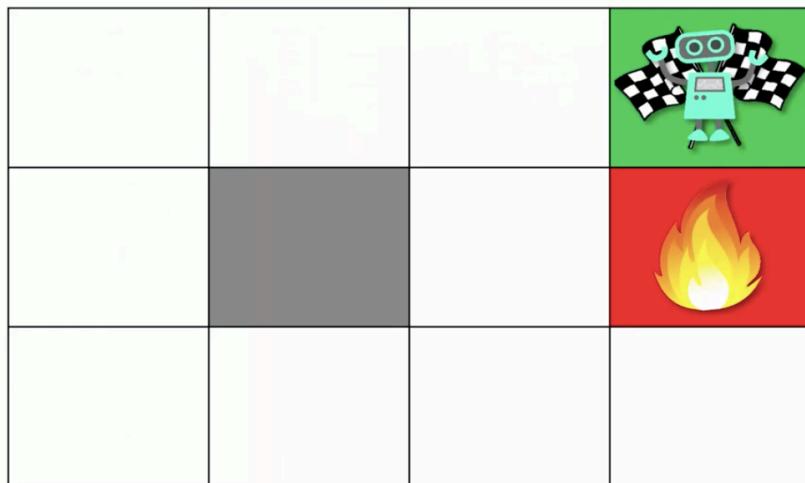
The Bellman Equation



Artificial Intelligence A-Z

© SuperDataScience

The Bellman Equation

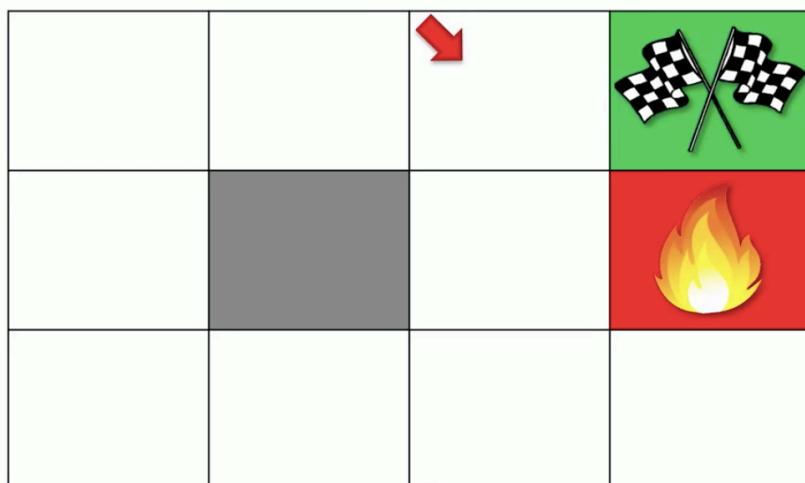


Artificial Intelligence A-Z

© SuperDataScience

Now it got a +1 reward. Now the agent starts to ask question that how did it get to here or what was the preceding state it was in and what action did I take to come to this square.

The Bellman Equation

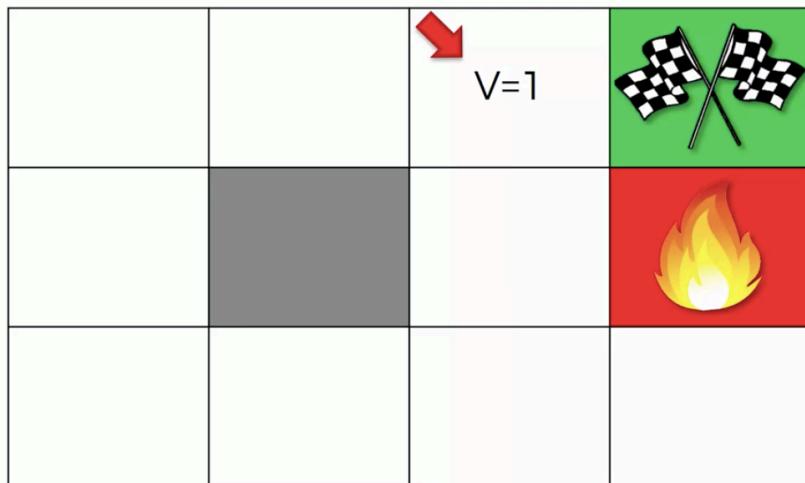


Artificial Intelligence A-Z

© SuperDataScience

Now the agent looks back to the preceding state and knows that whenever it is in the arrow, all it needs to do is go right.

The Bellman Equation

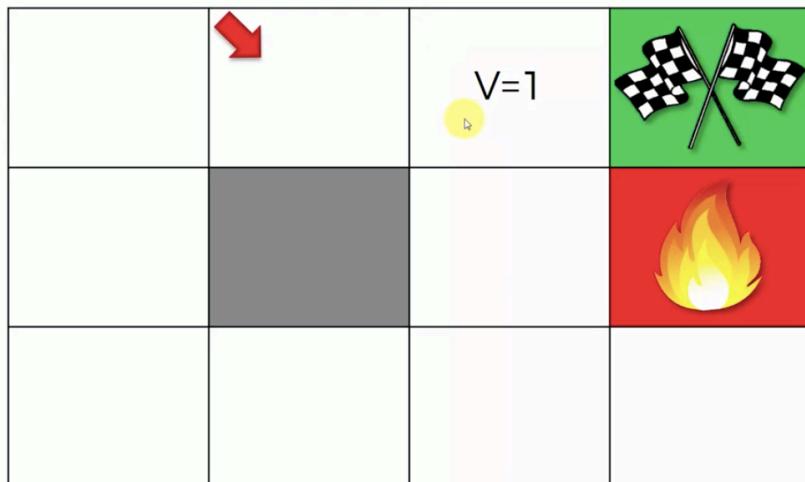


Artificial Intelligence A-Z

© SuperDataScience

Now the agent marks for itself value of 1 in the preceding state because it leads exactly to reward +1

The Bellman Equation

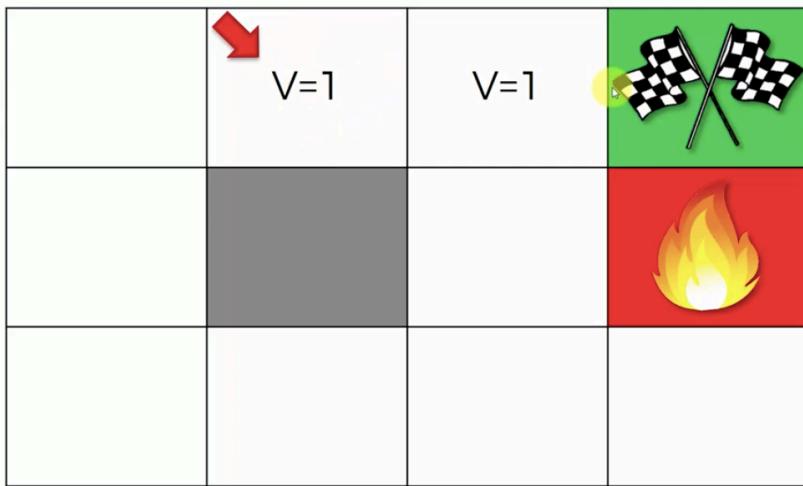


Artificial Intelligence A-Z

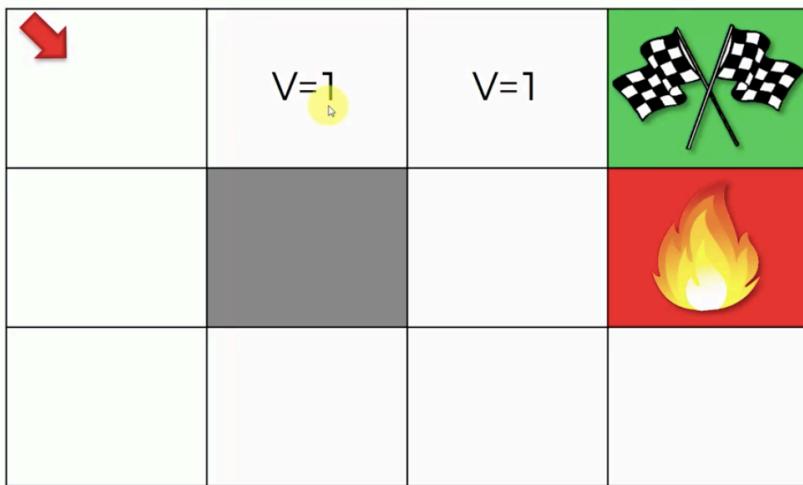
© SuperDataScience

Now agent considers 2 squares before it reached to the reward.

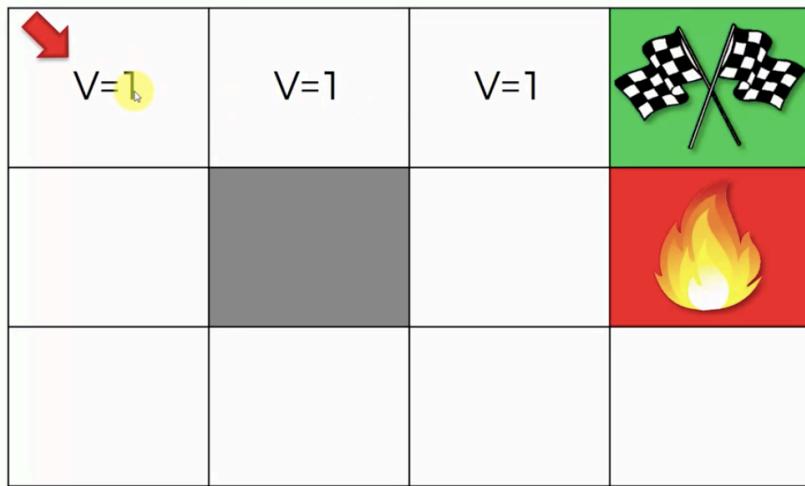
The Bellman Equation



The Bellman Equation



The Bellman Equation



The Bellman Equation



The Bellman Equation

$V=1$	$V=1$	$V=1$	
$V=1$			
 $V=1$			

Artificial Intelligence A-Z

© SuperDataScience

V is equal to 1 in all of the squares from the starting point.

The Bellman Equation

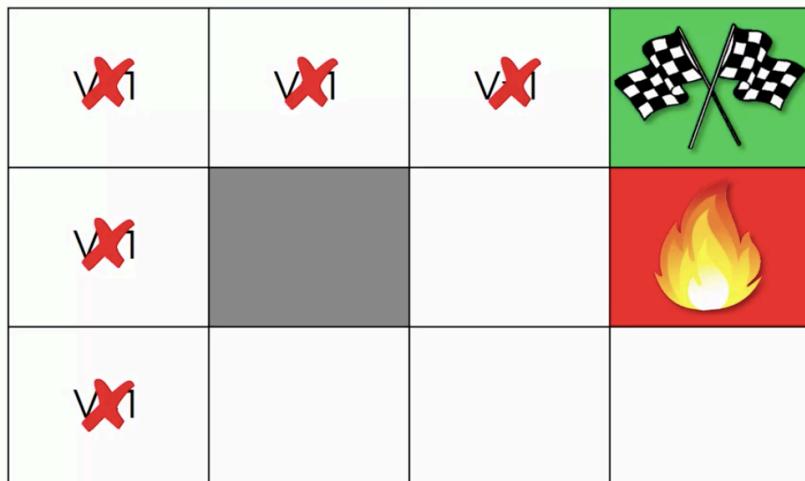
  	$V=1$	$V=1$	
$V=1$			
$V=1$			

Artificial Intelligence A-Z

© SuperDataScience

Now if after this we put the agent at this position that it has two $V=1$ in its right and two others below it. The agent cannot remember the way so..

The Bellman Equation



Artificial Intelligence A-Z

© SuperDataScience

This approach doesn't work

The Bellman Equation



Artificial Intelligence A-Z

© SuperDataScience

The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$

Artificial Intelligence A-Z

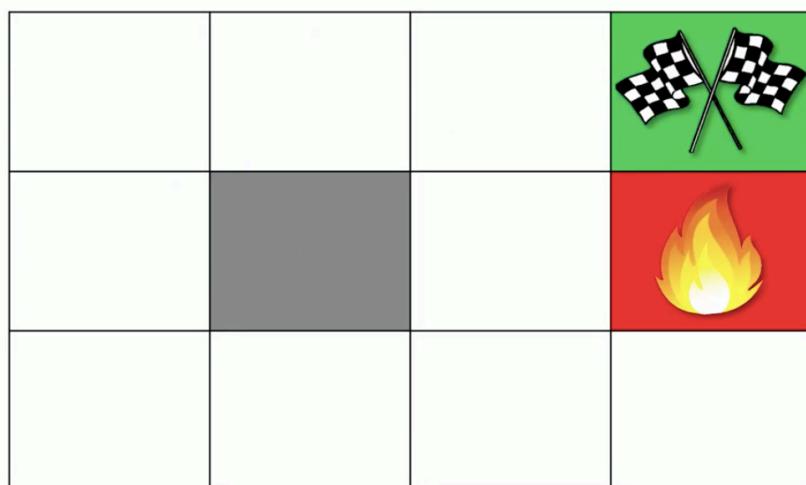
© SuperDataScience

This is Bellman Equation. V is the value of being in a certain state. S is current or given state. S prime is the following / next state.

When we go in a certain state we get an instant reward by R(s,a) that is -1 or 1 at the end of game and sometimes it can be 0 which is throughout the game as well. Gamma is there to solve the problem that we had before in which agent doesn't know in which way it should go. Gamma is also called discounting factor.

The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$

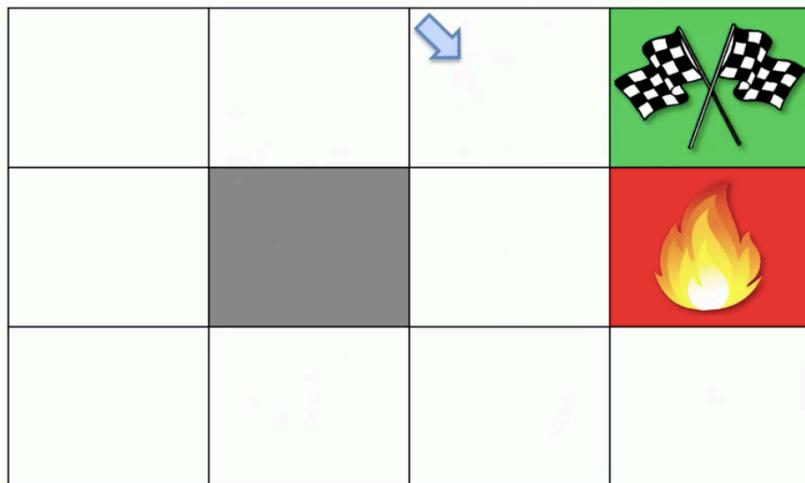


Artificial Intelligence A-Z

© SuperDataScience

The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$



Artificial Intelligence A-Z

© SuperDataScience

In here we will calculating the $R(s,a)+\gamma V(s')$ for one state before the the +1 reward which its max is one step to the right or in other word max is the direction it goes.

The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$



Artificial Intelligence A-Z

© SuperDataScience

The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$

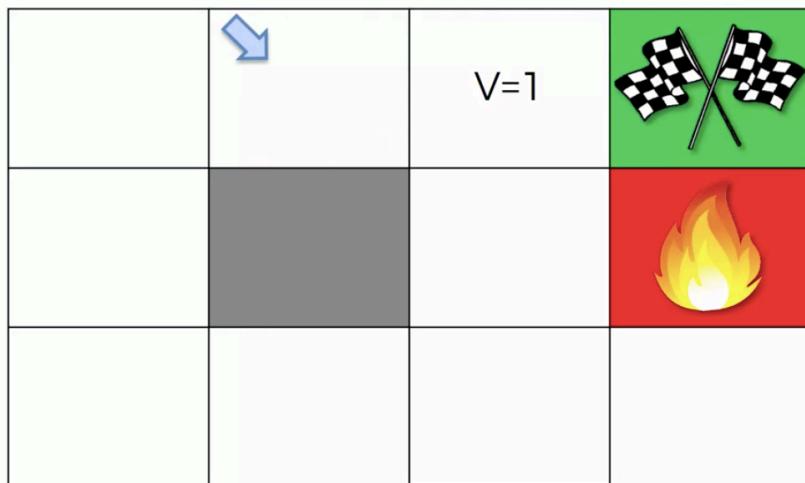


Artificial Intelligence A-Z

© SuperDataScience

The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$



Artificial Intelligence A-Z

© SuperDataScience

The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$

	 V=0.9	V=1	
			

$$\gamma = 0.9$$

The Bellman Equation

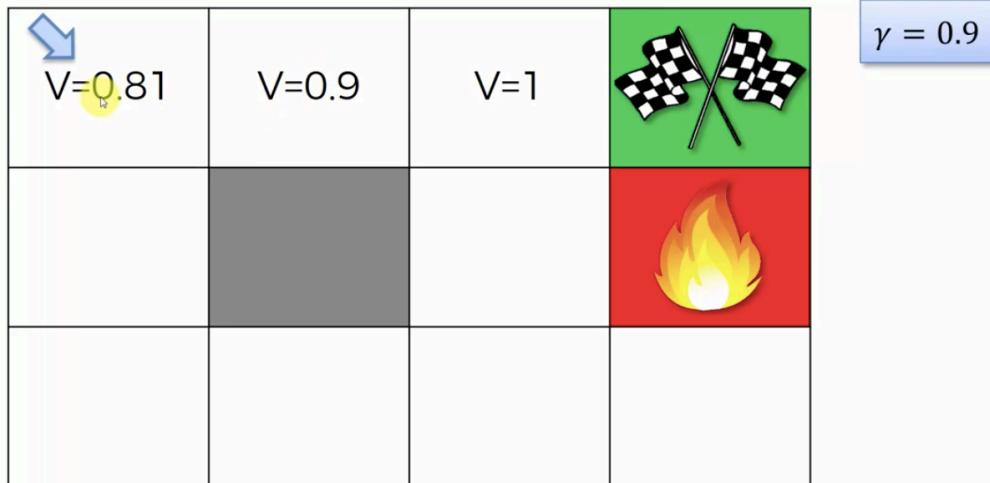
$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$

 	V=0.9	V=1	
			

$$\gamma = 0.9$$

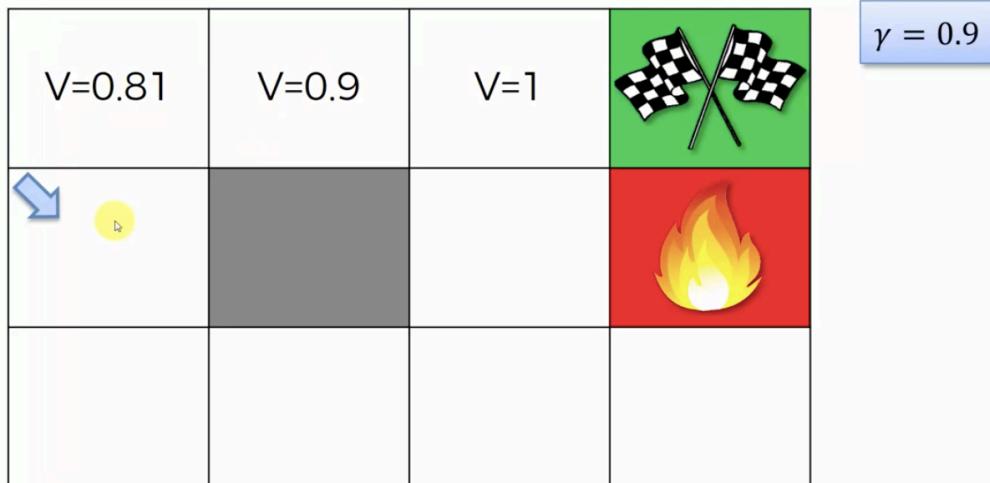
The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$



The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$



The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$

V=0.81	V=0.9	V=1		$\gamma = 0.9$
V=0.73				

Artificial Intelligence A-Z

© SuperDataScience

The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$

V=0.81	V=0.9	V=1		$\gamma = 0.9$
V=0.73				
V=0.66				

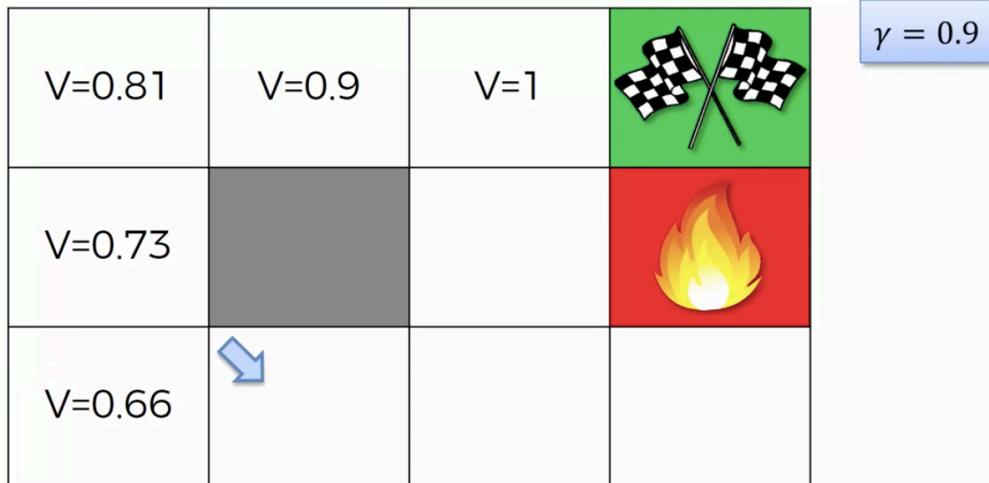
Artificial Intelligence A-Z

© SuperDataScience

If you noticed this works backward from reward +1 to the starting point

The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$

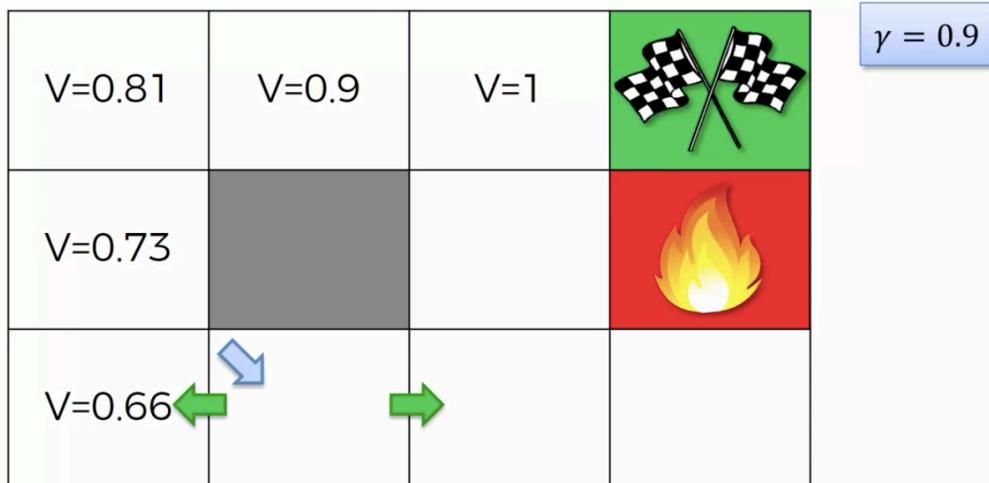


Artificial Intelligence A-Z

© SuperDataScience

The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$



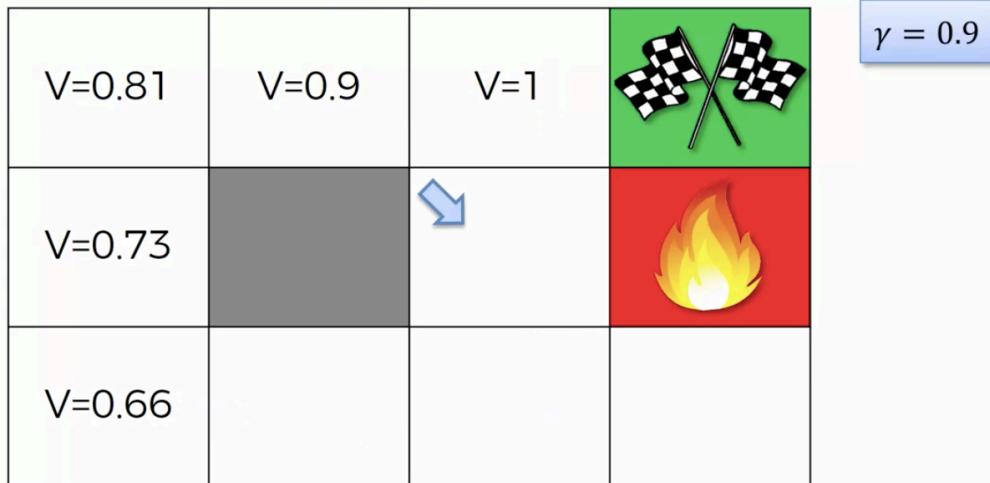
Artificial Intelligence A-Z

© SuperDataScience

We are not starting from here because there is a chance that this is a shorter path. So..

The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$



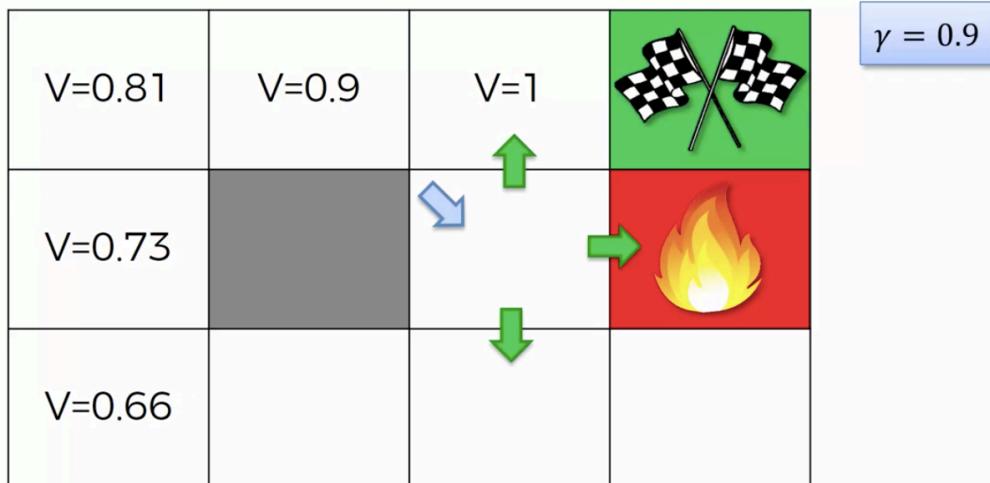
Artificial Intelligence A-Z

© SuperDataScience

we start backward from reward +1

The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$



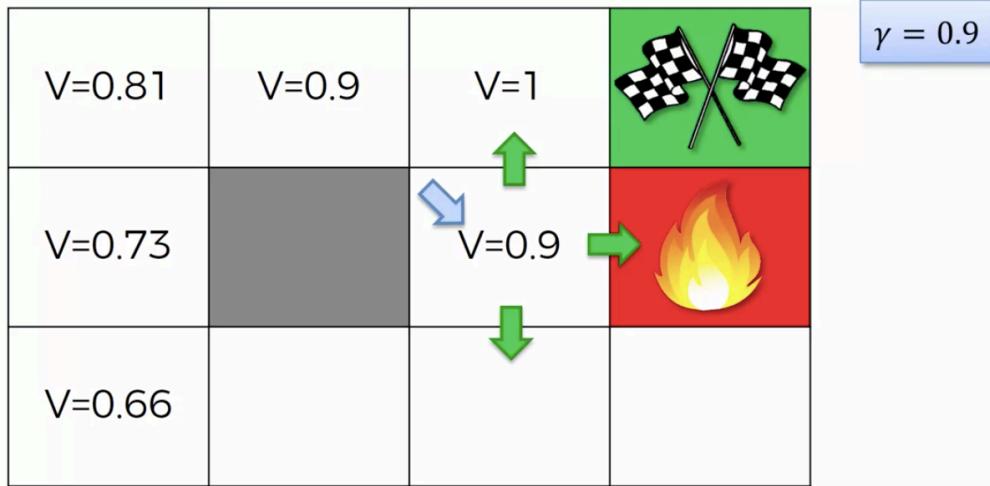
Artificial Intelligence A-Z

© SuperDataScience

In reality we have 4 possible action which in left action it bump into wall but in here for simplicity we say 3 possible action

The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$



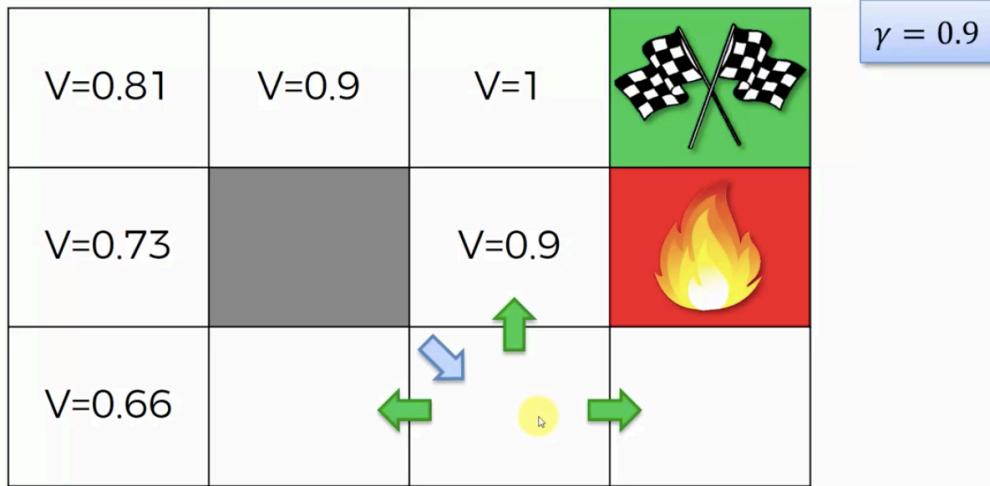
The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$



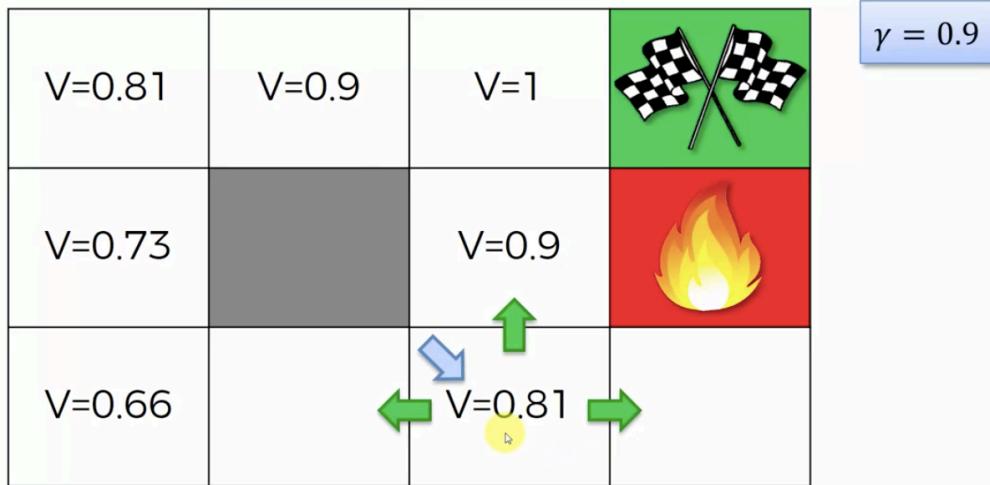
The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$



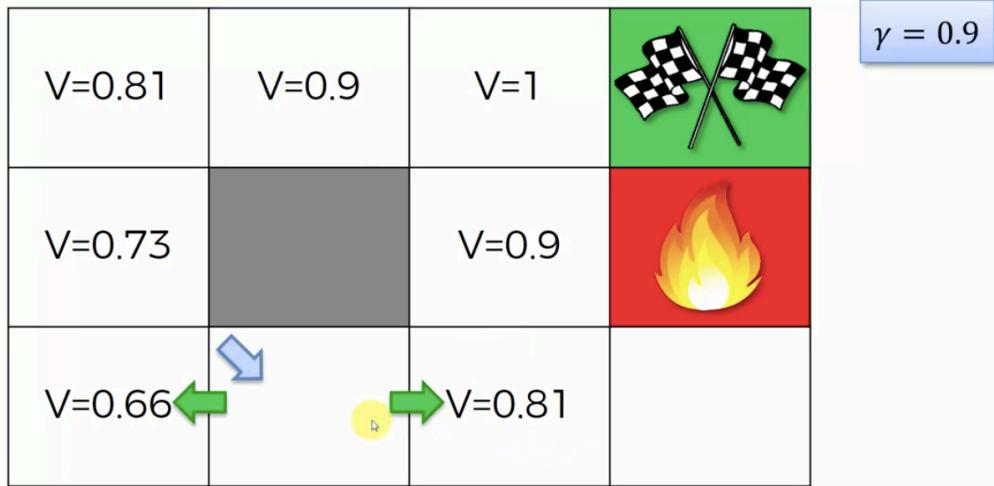
The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$



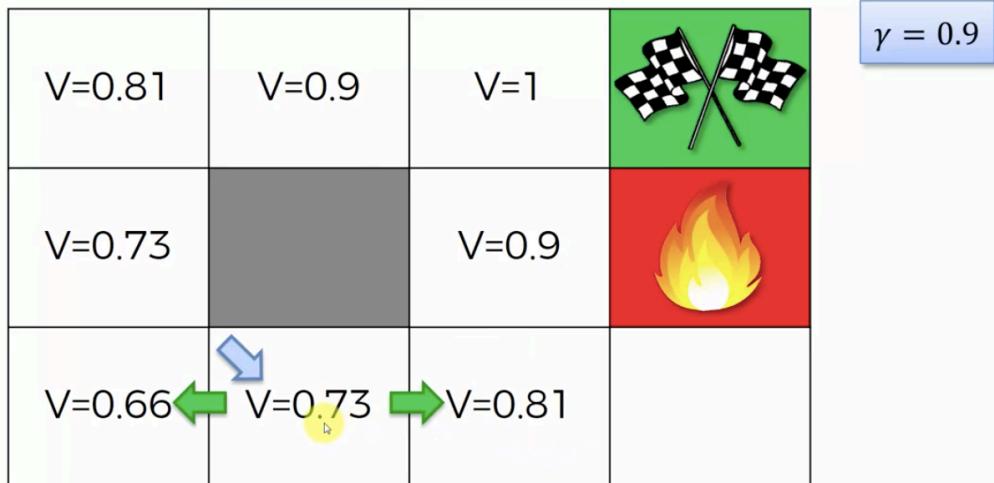
The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$



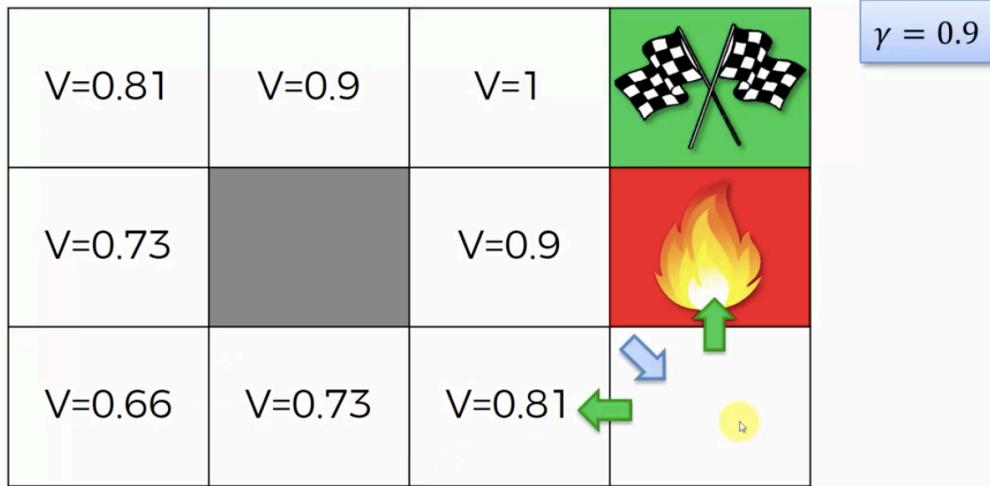
The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$



The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$



The Bellman Equation

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$

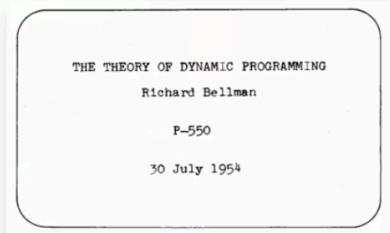


Additional Reading

Additional Reading:

The Theory of Dynamic Programming

By Richard Bellman (1954)



Link:

<https://www.rand.org/content/dam/rand/pubs/papers/2008/P550.pdf>