The brain behind the car:

Reinforcement learning:

**Reinforcement learning** (**RL**) is an area of [machine learning](https://en.wikipedia.org/wiki/Machine_learning) inspired by [behaviourist psychology](https://en.wikipedia.org/wiki/Behaviorism" \o "Behaviorism), concerned with how [software agents](https://en.wikipedia.org/wiki/Software_agent) ought to take [*actions*](https://en.wikipedia.org/wiki/Action_selection) in an *environment* so as to maximize some notion of cumulative *reward*. The problem, due to its generality, is studied in many other disciplines, such as [game theory](https://en.wikipedia.org/wiki/Game_theory), [control theory](https://en.wikipedia.org/wiki/Control_theory), [operations research](https://en.wikipedia.org/wiki/Operations_research), [information theory](https://en.wikipedia.org/wiki/Information_theory), [simulation-based optimization](https://en.wikipedia.org/wiki/Simulation-based_optimization), [multi-agent systems](https://en.wikipedia.org/wiki/Multi-agent_system), [swarm intelligence](https://en.wikipedia.org/wiki/Swarm_intelligence), [statistics](https://en.wikipedia.org/wiki/Statistics" \o "Statistics)and [genetic algorithms](https://en.wikipedia.org/wiki/Genetic_algorithm). In the operations research and control literature, reinforcement learning is called *approximate dynamic programming,* or *neuro-dynamic programming.* The problems of interest in reinforcement learning have also been studied in the [theory of optimal control](https://en.wikipedia.org/wiki/Optimal_control_theory), which is concerned mostly with the existence and characterization of optimal solutions, and algorithms for their exact computation, and less with learning or approximation, particularly in the absence of a mathematical model of the environment. In [economics](https://en.wikipedia.org/wiki/Economics) and [game theory](https://en.wikipedia.org/wiki/Game_theory), reinforcement learning may be used to explain how equilibrium may arise under [bounded rationality](https://en.wikipedia.org/wiki/Bounded_rationality).

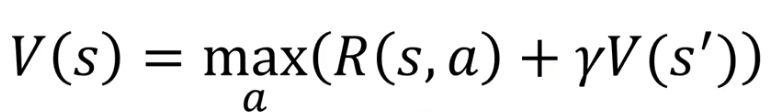
Reinforcement learning uses reward system technique. In the self driving car, each time the car moves in the desired direction a positive reward is obtained and when moved in the wrong direction or collided with an obstacle a negative reward is given. Just like training animals such as dogs by giving rewards and punishments. This car is first trained in the environment.

After the training the car gain from the experience and moves in a desired manner.

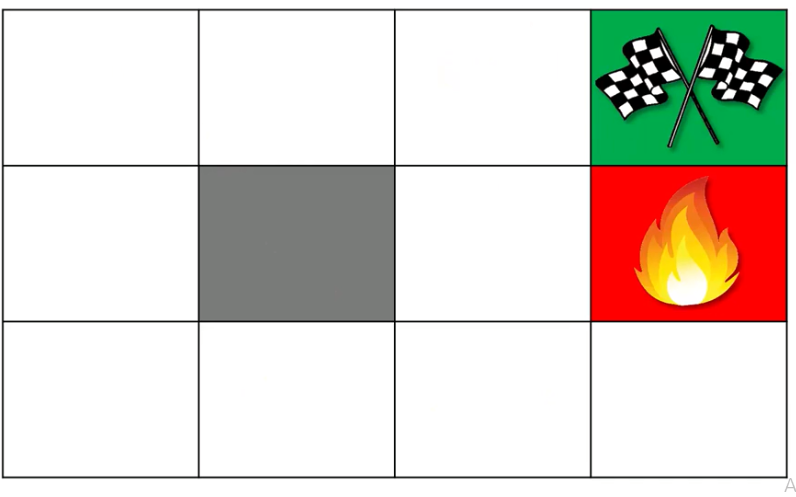
Bellman Equation:

The heart of reinforcement learning is Bellman equation. This equation monitors the reward system for the vehicle.

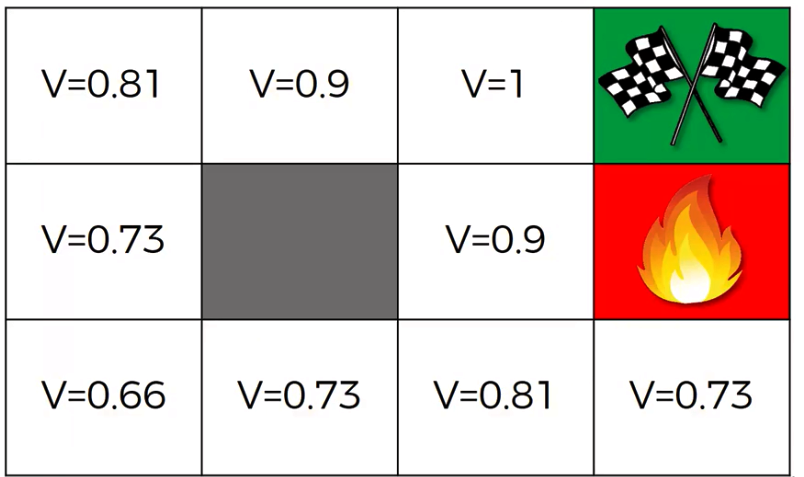
A **Bellman equation**, named after [Richard E. Bellman](https://en.wikipedia.org/wiki/Richard_E._Bellman), is a [necessary condition](https://en.wikipedia.org/wiki/Necessary_condition) for optimality associated with the mathematical [optimization](https://en.wikipedia.org/wiki/Optimization_(mathematics)) method known as [dynamic programming](https://en.wikipedia.org/wiki/Dynamic_programming). It writes the value of a decision problem at a certain point in time in terms of the payoff from some initial choices and the value of the remaining decision problem that results from those initial choices. This breaks a dynamic optimization problem into a [sequence](https://en.wikipedia.org/wiki/Sequence) of simpler subproblems, as [Bellman's “principle of optimality”](https://en.wikipedia.org/wiki/Bellman_equation#Bellman's_Principle_of_Optimality) prescribes.



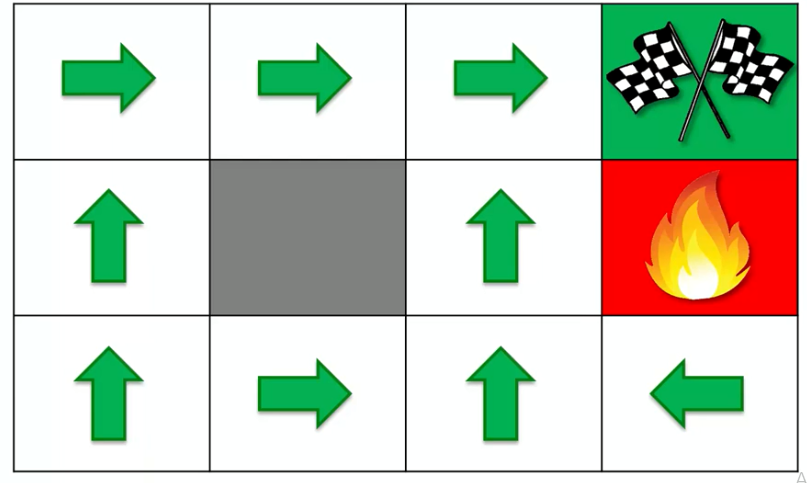
Here R(s,a) is the reward given in the next state. V(S’) is the value of next state. Gamma is the discount factor.



Consider the above maze. The goal is to reach the green point. If it falls into the fire pit the -1 reward is given. If it reaches the green mark then +1 reward is given.



If the value V is calculated from V=1 using the discount factor gamma equal to 0.9 the other values are calculated. Therefore now using these values a car or a bot can reach the goal using these values in an more efficient way. It tries to take the path that gives it maximum V value.



The above diagram shows the directions car would take in respective positions to reach the goal.