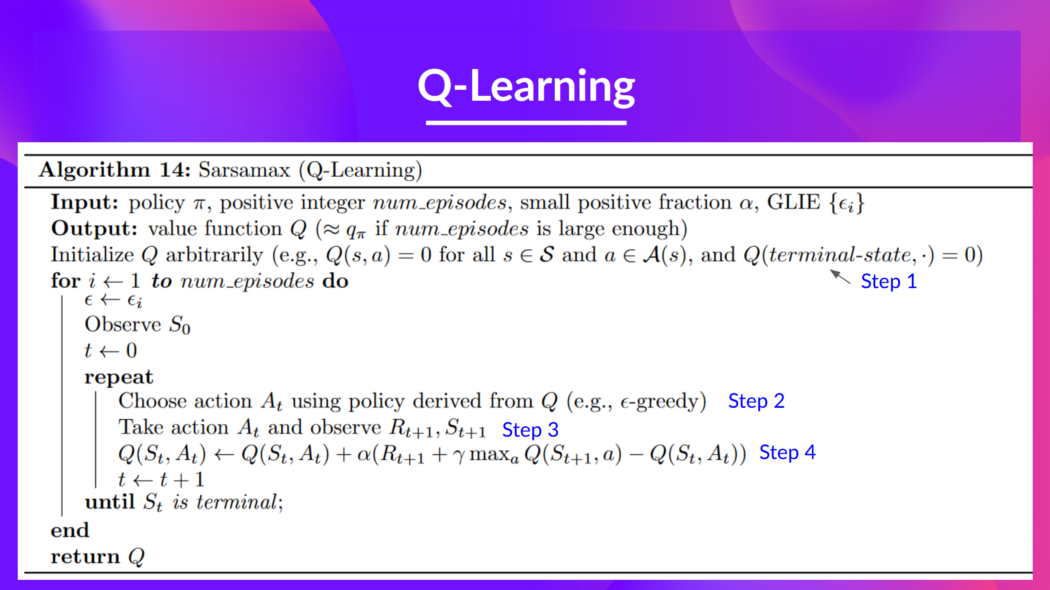
Q-Learning Algorithm

**On-policy vs Off-policy**:

1. On-policy: using the same policy for acting and updating.
2. Off-policy: using a different policy for acting (inference) and updating (training).

Q-Learning is an off-policy value-based method that uses a TD approach to train its action-value function:

* Value-based method: finds the optimal policy indirectly by training a value or action-value function that will tell us the value of each state or each state-action pair.
* TD approach: updates its action-value function at each step instead of at the end of the episode.



**Step 1**: Initialize the Q-table.

**Step 2**: Choose an action using the epsilon-greedy strategy.

The epsilon-greedy strategy is a policy that handles the exploration/exploitation trade-off.

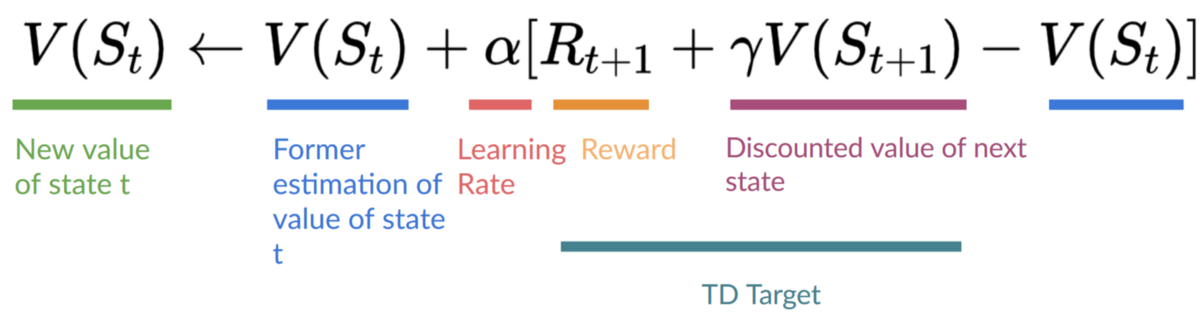
The idea is that, with an initial value of ɛ = 1.0:

* With probability 1 — ɛ : we do exploitation (aka our agent selects the action with the highest state-action pair value).
* With probability ɛ: we do exploration (trying random action).

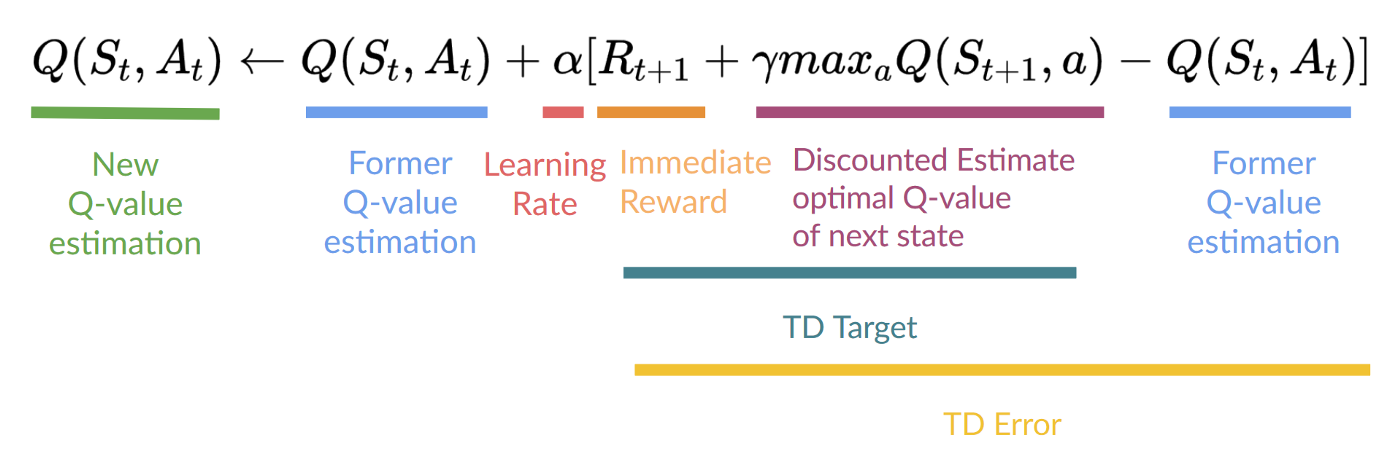
At the beginning of the training, the probability of doing exploration will be huge since ɛ is very high, so most of the time, we will explore. But as the training goes on, and consequently our Q-table gets better and better in its estimations, we progressively reduce the epsilon value since we will need less and less exploration and more exploitation.

**Step 3**: Perform action At, get reward Rt+1 and next state St+1.

**Step 4**: Update Q(St, At)



But since we are using greedy policy to update Q(St, At), the expression changes to:



How do we form the TD target?

1. We obtain the reward after taking the action Rt+1
2. To get the best state-action pair value for the next state, we use a greedy policy to select the next best action. Note that this is not an epsilon-greedy policy, this will always take the action with the highest state-action value.