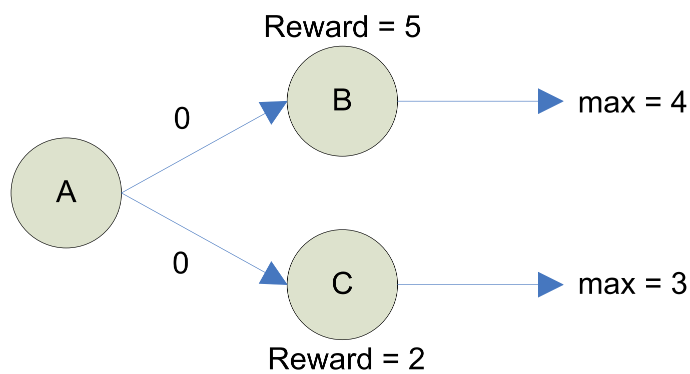
Q Learning

Problem: The following state diagram describes a learner's current knowledge about its environment:



Note that nothing is yet known about moving from state A to states B or C; i.e. the stored estimate of those values is 0. The two max values represent the stored estimates of the best possible courses of action when in states B or C; i.e. the expected reward for choosing the best possible action when in states B or C.

The Q-Learning algorithm describes how stored estimates are updated after taking a particular action (exploring).

<u>Q-Learning Algorithm:</u>

Initialize all value estimates Q(s_t, a_t)
For each "episode":
  Initialize state for current time s_t
  Repeat:
    Choose action a_t based on current policy π
    Observe reward r_{t+1} and move to new state s_{t+1}
    Update value estimate:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \eta[r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

    Make new state s_{t+1} be the current state s_t
  Until reaching a terminal state

In the algorithm, Q(s, a) is the hypothesized value of the "goodness" of taking action a while in state s. The parameter r is the immediate reward for taking a particular action. The value Q(s_{t+1}, a_{t+1}) represents the current estimate of the value of the most advantageous *next* state/action pair. As usual, η is the learning factor, which is gradually reduced as value estimates converge, and γ is the discount factor, a parameter controlling the importance of history.

Assume that η, the learning factor, is currently set to 0.6; and that $\gamma = 0.5$ is being used as the discount factor.

Executing the "exploratory" phase of the algorithm, when taking the action that moves from state A to either B or C, would result in the following update to the stored estimates of the value of either of those actions.

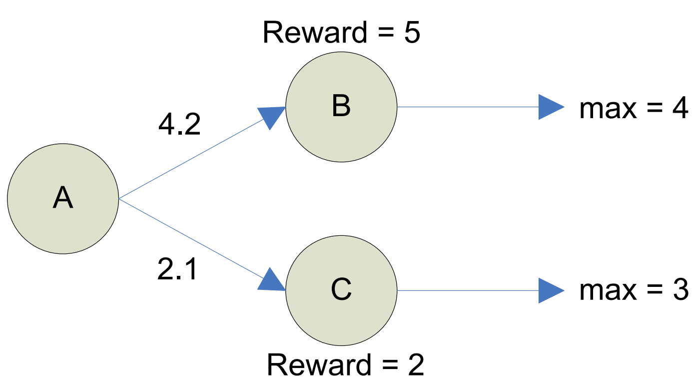
A → B
Upon moving to state B, the stored value for taking the action that moves from state A to state B, is updated:

$$Q(\text{state-A, action-B}) = 0 + 0.6\ (5 + 0.5 * 4 - 0) = 4.2$$

A → C
Upon moving to state C, the stored value for taking the action that moves from state A to state C, is updated:

$$Q(\text{state-A, action-C}) = 0 + 0.6\ (2 + 0.5 * 3 - 0) = 2.1$$

The model is then updated to reflect the new information about the environment:



As the process of *exploration* continues over time, the stored estimates of the value of each action are continuously updated, and probabilistically converge towards an optimal policy. This facilitates the shift to *exploit* mode, in which the action expected to produce the maximum cumulative reward is always chosen.