Tutorial #13

December 9, 2021 12:58 PM

Deep Q-learning

- Recall the Bellman equations for

  (i) Value function :

$$V(s) = \max_{a \in \mathcal{A}} \mathbb{E}\left[r(a,s,s') + \gamma V(s') \right]$$

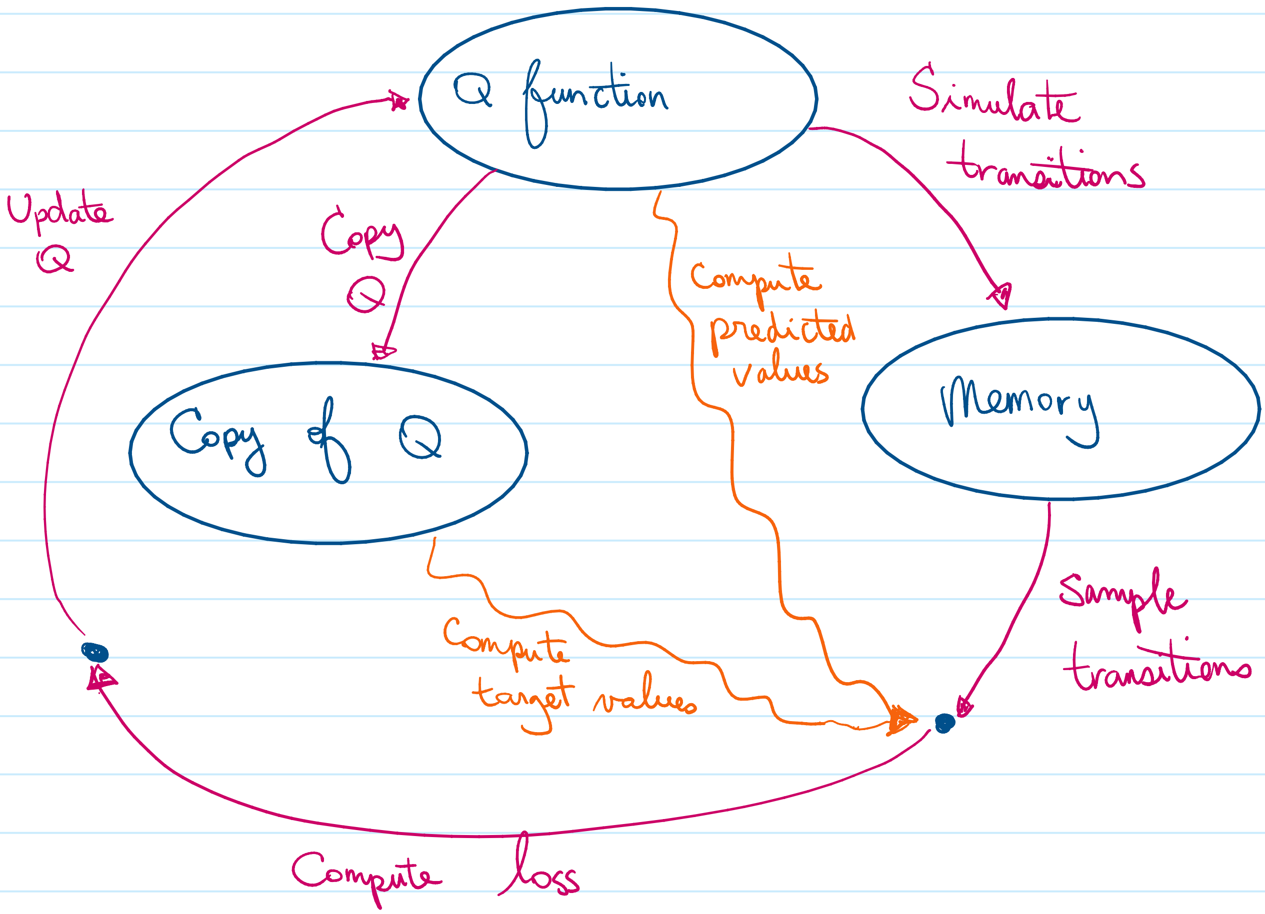
  (ii) Q function :

$$Q(s,a) = \mathbb{E}\left[r(a,s,s') + \gamma V(s') \right]$$

$$= \mathbb{E}\left[r(a,s,s') + \gamma \max_{a' \in \mathcal{A}'} Q(s',a') \right].$$

- We want to use the Q-function (as a neural network) to find the best policy.

Representation of the algorithm



More details on deep Q-learning

- Q function (main and target networks):

  ANN with state dimension as inputs and number of actions as outputs, to give

  $$Q(s, a^{(1)}),\ Q(s, a^{(2)}),\ \ldots,\ Q(s, a^{(|\mathcal{A}|)}).$$

- Memory: a replay buffer with a fixed size to store & sample transitions
  (s_t, a_t, r_t, s_{t+1})

- Simulate transitions:

  ε-greedy policy, where we select the best action (with respect to the Q function) with probability $1-\varepsilon$, and a random action with probability ε. It allows exploration & exploitation.

- Predicted Q value: $Q(s_t, a_t)$

- Target Q value: $r_t + \gamma \max_{a' \in \mathcal{A}} Q(s_{t+1}, a')$

- Update loss: Mean squared error between the predicted and target values.

- Other implementation details:

  (i) frequence of the copy of Q as the target network.
  (ii) decay the ε in the ε-greedy.
  (iii) mini-batch size when sampling transitions from the memory.
  (iv) select only valid actions.