Reinforcement Learning (RL)

Introduction

- The concept *of reinforcement learning* incorporates an *agent* that solves the problem in hand by *interacting with its environment*.
- *Agent* learns what *action* to take at what situation from signals that *agent* receives as a *response* from its *environment* to *actions* taken at a specific situation of the environment.
- *Agent* seeks to *achieve its goal* by interacting with its environment and receiving response signals to actions it takes.
- Interaction with its environment (i.e., experience) provides *agent* with the *accumulated knowledge* of whether or not (or to what extent) the environment likes the actions taken by the agent.

Reinforcement Learning

- Two actors
  - Learner, called the *agent*
  - *Environment*, the domain providing feedback to agent so it optimizes its actions.
- Agent *learns from interaction* with an unknown *environment*
- Agent's interaction consists of responses emitted by the environment against actions it takes at every time instant.

Reinforcement Learning Loop

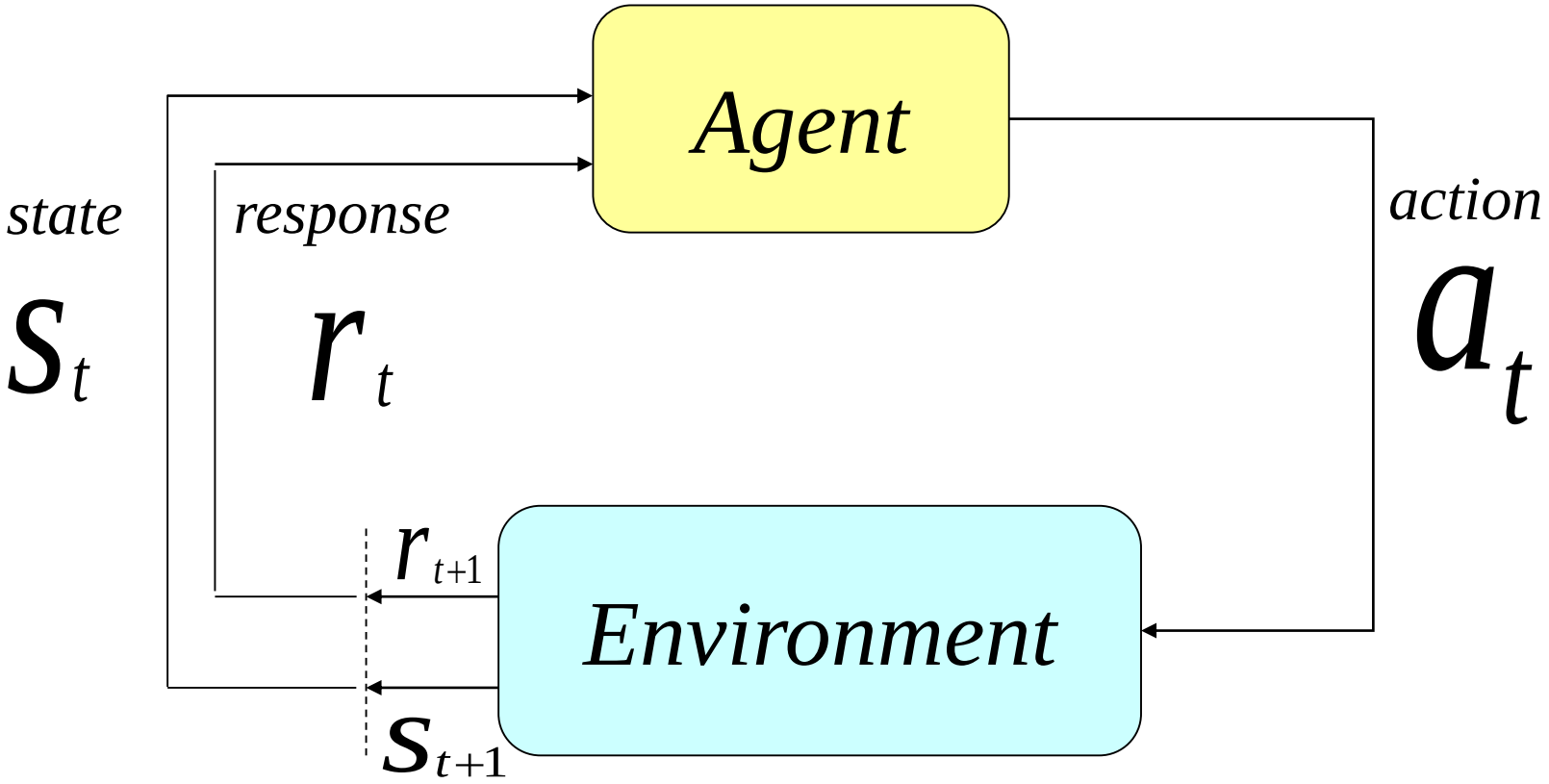

Figure reproduced from the figure on page 52 in reference [1]

Elements of RL

- *Policy* or rule set of agent behavior
- *Response* or reward function
- *Value* function
- *Model* of the environment

Elements of RL ...(2)

- *Policy:* Agent's way of behaving at a given time. A little more formally: *a mapping from perceived states to actions*. Analogy with psychology: *stimulus-response* rules.

- *Response* or reward function

Elements of RL ...(3)

- *Value functions* are functions of states ($V^{\pi}(s)$) or of state-action pairs ($Q^{\pi}(s,a)$) that *define a basis for agent's decision* about "how good" it is to be at a state or to select an action at a specific state, respectively, for a specific policy $\boldsymbol{\pi}$.

Elements of RL ...(4)

- *Model of the environment* *is a representative for the behavior of the environment*.
- Using a model of the environment the agent operates in, the resultant next state and next reward may be estimated.
- This information may be further used for *planning*. That is, the agent may form a structure of decisions considering possible future situations before they are actually experienced.

Examples to RL: Tavla

- The way we learn how to play tavla (backgammon)
  - the first times we play the game we just apply the rules of the game and observe how the game proceeds. (Repetitions or "*episodes*" of the game)
  - As we start to associate situations and actions in these situations with a good or bad end of the game we implicitly assign *values* to these situations and actions we take in these situations about whether or not or how "*favorable*" or "*unfavorable*" they are;
  - if we are experienced enough (i.e., after sufficiently many games) we start *planning* or building *strategies*.

Examples to RL: A Weekly Routine

- You are a soccer player and getting ready for your soccer game.
  - Following your last class that day, you are heading for the locker room. At the desk, you ask the receptionist for a locker key. You get the key, write your name, leave your student ID and go to the locker whose number matches with that on the key. You check to see and hopefully open the locker door.
  - You change your dresses and put on your soccer short and t-shirt. You hang your regular outfit and put your sport bag into the locker. You place the key in a secure pocket.
  - etc.
  - As you feel these routine activities are composed of many conditional behaviors and intermingled subgoal and goal relationships
  - For more examples you may check page 6 in reference [1].

An Extended Example: Tic-Tac-Toe

- Goal in tic-tac-toe is to
  - place your three marks in a row horizontally, vertically, or diagonally before your opponent does his/her.

- A draw: if no player has three in a row when the board is full.

Tic-Tac-Toe...(2)

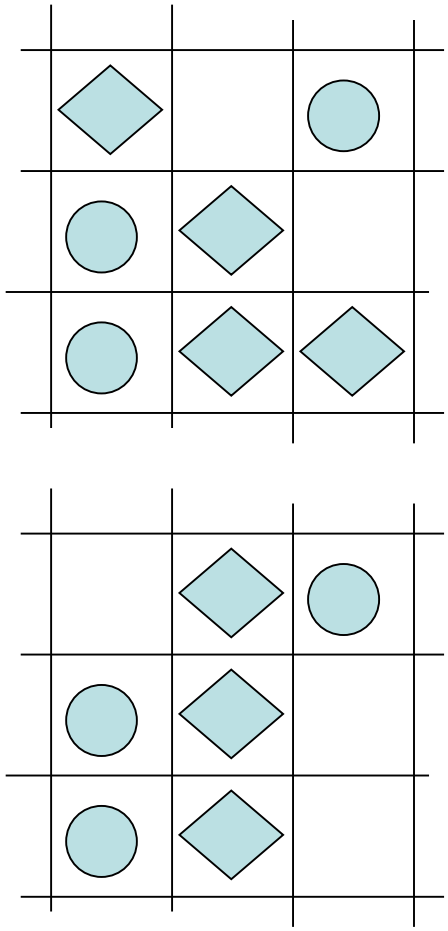

Figure 1: Two winning situations for diamond

- *How to define states of RL problem for this game*:
  - *States*: each situation of the board is a state.
  - Each state where your three marks are in a row is a *goal state*.
  - Each state where the opponent's three marks are in a row is a state you cannot win from.

Tic-Tac-Toe...(3)

- ***How to set up a RL problem for this game?..***
  - Each state is assigned a number to represent the *value* of the state.
  - Assume a *table of numbers*, each number representing the value of one state where the number is an *estimate for the probability of winning from that state*.
  - The *table is the learned state-value function*.
  - A *better* state is one with a *higher current estimate of probability of winning from this state than with that from another state*.

Tic-Tac-Toe...(4)

- ***Initialization of state values:***
  - We assign 1 to our goal states since the probability of winning from our goal states is 1 (i.e., at these states we have already won.)
  - Similarly 0 is assigned to the opponent's goal states since it is improbable for us to win from these states (i.e., we have already lost the game.)
  - All other states we may initially assign 0.5 to mean we may lose or win equally probably from these states.

Tic-Tac-Toe...(5)

- ***How to decide on which state to move to/what action to take?***
  - Whatever state we are at we would like to approach one of our goal states and keep away from the opponent's goal states at the same time. *Why?*
  - The goal state(s) are the ones with the highest values; hence from a neighbor state to a goal state we should select a move that leads to the goal state.
  - In the same fashion, from any state we should select a move that takes us to the state with the highest value. *Why?*

The answer to both questions is we would like to move to or be at an accessible state with one of the highest possible values since to move to or to be at a higher value state makes it more probable for us to win the game.

Tic-Tac-Toe...(6)

- ***How to decide on what state to move to?... Cont'd***
  - the move towards the accessible state(s) with the highest value is called a *greedy* move.
  - if we select to move, in contrast with this, to a state with an intermediate (i.e., not the highest) value, meaning to give an effort to find a state that is a candidate to have a higher value in the future than the state with the highest value which would not otherwise be revealed since its current value is not high due to the occurrence of what is less likely to happen. This type of move is called an *exploratory* move.

Tic-Tac-Toe...(7)

- ***How to update state values?... Cont'd***
- As we move to the state (or one of the states) with highest possible value that is accessible, we update the value of the source state (i.e., the state we departed) such that it is now closer to the value of the destination state (i.e., the state we arrived) typically using the following learning rule:

$$V(s) \leftarrow V(s) + \alpha\left[V(s') - V(s)\right]$$

where *s* and *s'* are the source and destination states, respectively, *V(.)* is the value function, and α is the learning rate parameter that specifies the contribution of the current update to the entirety of the state value. The above learning rule is called the *temporal-difference learning method*.

References

- [1] Sutton, R. S. and Barto A. G., "*Reinforcement Learning: An introduction*," MIT Press, 1998