



Reinforcement Learning

Introduction

U Kang
Seoul National University



In This Lecture

- Introduction to RL
- Examples and Elements of RL
- Example: Tic-Tac-Toe



Outline

- ➔ **Reinforcement Learning**
- Basic Examples of RL
- Elements of RL
- Extended Example: Tic-Tac-Toe
- Conclusion



Reinforcement Learning (RL)

- RL: learning from interaction
 - Learning what to do (how to map situations to actions) so as to maximize a numerical reward signal
- 2 distinguishing features
 - Trial and error
 - The learner is not told which actions to take, but instead must discover which actions yield the most reward by trying
 - Delayed reward
 - Actions may affect not only the immediate reward, but also the next situation and all subsequent rewards



RL vs. Supervised Learning

- Supervised learning
 - Learning from a training set of labeled examples
 - Goal: generalize its responses so that it acts correctly in situations not present in the training set
- RL
 - In interactive problems it is often impractical to obtain examples of desired behavior that are both correct and representative of all the situations in which the agent has to act
 - An agent must be able to learn from its own experience



RL vs. Unsupervised Learning

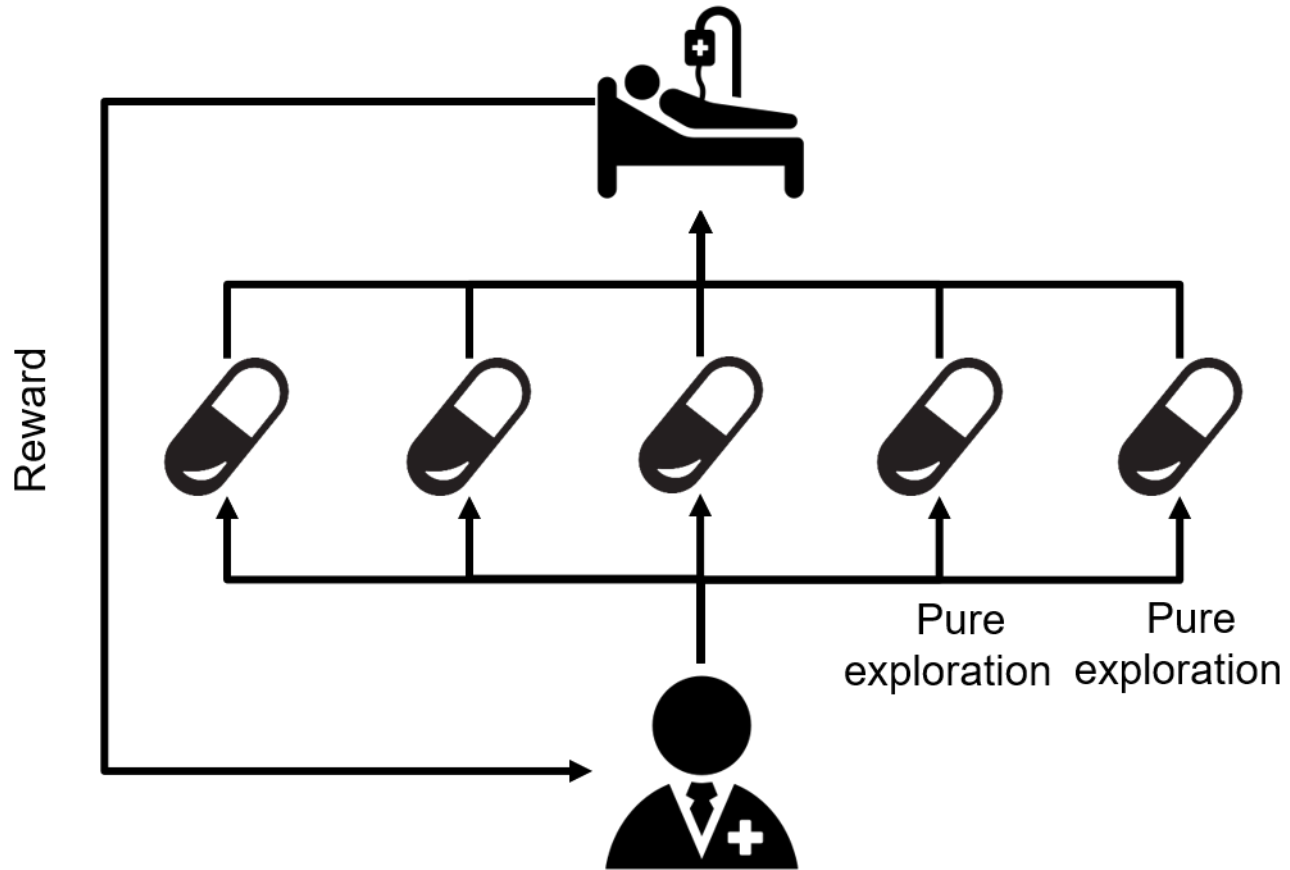
- Unsupervised learning
 - Finding structure hidden in collections of unlabeled data

- RL
 - RL tries to maximize a reward signal instead of trying to find hidden structures



Exploration vs. Exploitation

- Trade-off between exploration and exploitation
- Exploitation: to obtain a lot of reward, a RL agent must prefer actions that it has tried in the past and found to be effective in producing reward
- Exploration: to discover such actions, it has to try actions that it has not selected before
- Dilemma: neither exploration nor exploitation can be pursued exclusively without failing at the task. The agent must try a variety of actions and progressively favor those that appear to be best.
- This issue does not arise in supervised and unsupervised learning





Goal-Directed Agent

- RL considers the whole problem of a goal-directed agent interacting with an uncertain environment
- In contrast, most ML approaches consider isolated subproblems without addressing how they might fit into a larger picture



Interactions with Other Disciplines

- RL is part of a decades-long trend within artificial intelligence and machine learning toward greater integration with statistics, optimization, and other mathematical subjects
- RL with parameterized approximators addresses the classical “curse of dimensionality” in operations research and control theory.
- RL has interacted strongly with psychology and neuroscience, with substantial benefits going both ways.
 - Many of the core algorithms of RL were originally inspired by biological learning systems.
 - RL has also given back, both through a psychological model of animal learning that better matches some of the empirical data, and through an influential model of parts of the brain’s reward system.



Seeking General Principles

- RL is also part of a larger trend in AI back toward simple general principles.
- Strong methods
 - Since the late 1960's, many artificial intelligence researchers presumed that intelligence is due to the possession of a vast number of "knowledge": special purpose tricks, procedures, and heuristics.
- Weak methods
 - Seek simple and general principles of learning, search, and decision making
 - RL belongs to this category



Outline

Reinforcement Learning

 **Basic Examples of RL**

Elements of RL

Extended Example: Tic-Tac-Toe

Conclusion



Examples

- Chess
 - A move in a chess is informed both by planning (anticipating possible replies and counterreplies) and by immediate, intuitive judgments of the desirability of particular positions and moves
- Operation in petroleum refinery
 - An adaptive controller adjusts parameters of a petroleum refinery's operation in real time. The controller optimizes the yield/cost/quality trade-off on the basis of specified marginal costs without sticking strictly to the set points originally suggested by engineers.



Examples

- Learning to run
 - A gazelle calf struggles to its feet minutes after being born. Half an hour later it is running at 20 miles per hour
- Cleaning robot
 - A robot decides whether it should enter a new room in search of more trash to collect, or start trying to find its way back to its battery recharging station
 - It makes its decision based on the current charge level of its battery and how quickly and easily it has been able to find the recharger in the past



Examples

- Preparing breakfast
 - Related to a complex web of conditional behavior and interlocking goal–subgoal relationships
 - Walking to the cupboard, opening it, selecting a cereal box, then reaching for, grasping, and retrieving the box, ...
 - Each step involves a series of eye movements to obtain information and to guide reaching and locomotion
 - Rapid judgments are continually made
 - Each step is guided by goals, such as grasping a spoon or getting to the refrigerator, and is in service of other goals, such as having the spoon to eat with once the cereal is prepared and ultimately obtaining nourishment



Common Features

■ Interaction

- ❑ An active decision-making agent seeks to achieve a goal despite uncertainty about its environment.
- ❑ The agent's actions are permitted to affect the future state of the environment (e.g., the next chess position, the level of reservoirs of the refinery, the robot's next location and the future charge level of its battery), thereby affecting the actions and opportunities at later times
- ❑ Correct choice requires taking into account indirect, delayed consequences of actions, and thus may require foresight or planning



Common Features

- Uncertainty
 - The effects of actions cannot be fully predicted; thus the agent must monitor its environment frequently and react appropriately.
 - Tries to avoid overflowing when pouring milk into a bowl
 - Involve goals that are explicit in the sense that the agent can judge progress toward its goal based on what it can sense directly
 - The chess player knows whether or not he wins, the refinery controller knows how much petroleum is being produced, the gazelle calf knows when it falls, the cleaning robot knows when its batteries run down, ...




Common Features

- Improve performance over time
 - The chess player refines the intuition he uses to evaluate positions, thereby improving his play
 - The gazelle calf improves the efficiency with which it can run
 - A person learns to streamline making his breakfast
 - The knowledge the agent brings to the task at the start (either from previous experience with related tasks or built into it by design or evolution) influences what is useful or easy to learn, but interaction with the environment is essential for adjusting behavior to exploit specific features of the task



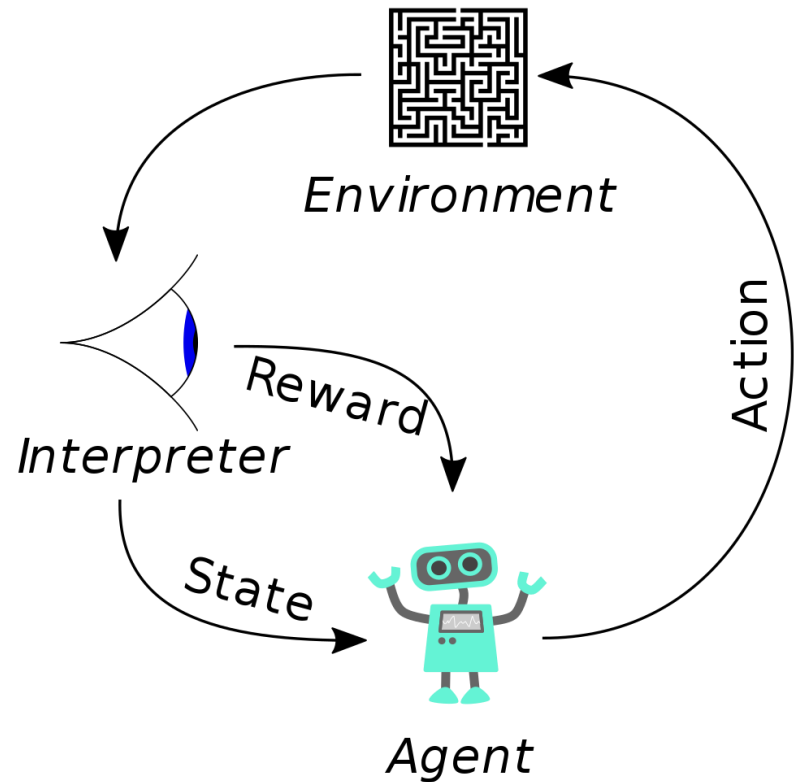
Outline

- Reinforcement Learning
- Basic Examples of RL
-  **Elements of RL**
- Extended Example: Tic-Tac-Toe
- Conclusion



Main Elements of RL

- Policy
- Reward signal
- Value function
- Model



[en.wikipedia.org/wiki/Reinforcement_learning#:~:text=Reinforcement%20learning%20\(RL\)%20is%20an,supervised%20learning%20and%20unsupervised%20learning.](https://en.wikipedia.org/wiki/Reinforcement_learning#:~:text=Reinforcement%20learning%20(RL)%20is%20an,supervised%20learning%20and%20unsupervised%20learning.)



Policy

- A mapping from state to action: defines the learning agent's way of behaving at a given time
- Corresponds to stimulus–response rules or associations in psychology
- Policy
 - may be a simple function or lookup table
 - may involve extensive computation such as a search process
- Policy is the core of a RL agent since it alone is sufficient to determine behavior
- In general, policies may be stochastic, specifying probabilities for each action



Reward Signal

- Defines the goal of a RL problem
- At each time step, the environment sends to the RL agent a reward
- Agent's goal: maximize the total reward it receives over the long run
- Biological system: reward ~ experiences of pleasure or pain
- Reward signal is the primary basis for altering the policy; if an action selected by the policy is followed by low reward, then the policy may be changed to select other action in that situation in the future
- Reward signals may be stochastic functions of the state of the environment and the actions taken



Value Function

- Value function: state \rightarrow value
- Value of a state s : total amount of reward to accumulate over the future, starting from s .
- Value vs. reward: reward signal indicates what is good in an immediate sense; a value function specifies what is good in the long run, considering the states that are likely to follow and the rewards in those states
 - A state might always yield a low immediate reward but still have a high value because it is regularly followed by other states that yield high rewards



Value Function

- Rewards are in a sense primary, whereas values, as predictions of rewards, are secondary. Without rewards there could be no values
- Nevertheless, it is values with which we are most concerned when making and evaluating decisions. We seek actions that bring about states of highest value
- It is much harder to determine values than it is to determine rewards, since they must be estimated and re-estimated from the sequences of observations an agent makes over its entire lifetime.
- Efficiently estimating values is the most important part in almost all RL algorithms



Model of Environment

- The model mimics the behavior of the environment, or more generally, allows inferences to be made about how the environment will behave
 - E.g., given a state and action, the model might predict the resultant next state and next reward
- Models are used for planning, by which we mean any way of deciding on a course of action by considering possible future situations before they are actually experienced
- Methods for solving RL problems
 - Model-based methods: use models
 - Model-free methods: trial-and-error learners (opposite of planning)
 - Some methods both 1) learn a model and use it for planning, and 2) learn by trial and error as well



RL vs. Evolutionary Methods

- Evolutionary methods
 - E.g., genetic algorithms
 - Apply multiple static policies each interacting over an extended period of time with a separate instance of the environment.
 - The policies that obtain the most reward, and random variations of them, are carried over to the next generation of policies, and the process repeats
 - Useful when 1) space of policies is small, or 2) can be structured so that good policies are common or easy to find



RL vs. Evolutionary Methods

■ RL


- Learn while interacting with the environment
- Search for best policy (a function from state to action)
- Uses the states and actions an agent experiences

■ Evolutionary methods

- Do not learn with interaction (no value function)
- Do not exploit the fact that policy is a function from state to action
- Do not use the information of states and actions

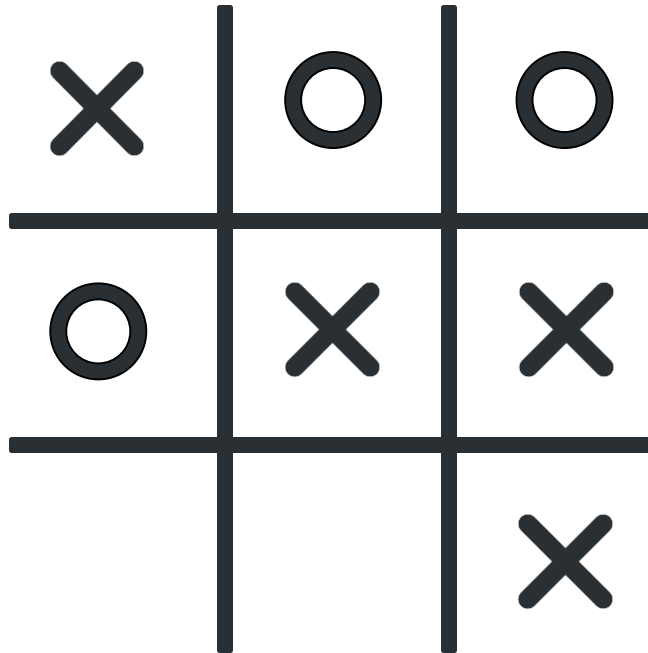


Outline

- Reinforcement Learning
- Basic Examples of RL
- Elements of RL
-  **Extended Example: Tic-Tac-Toe**
- Conclusion



Tic-Tac-Toe





Solving Tic-Tac-Toe via RL

- Set up value function
 - Set up a “value” table of numbers, one for each possible state; each number represents the probability of winning from that state
 - State A has higher value than state B if $P(\text{win from A}) > P(\text{win from B})$
 - If we play Xs, then for all states with three Xs in a row the probability of winning is 1; for all states with three Os in a row, the probability is 0
 - We set the initial values of all the other states to 0.5, representing a guess that we have a 50% chance of winning



Solving Tic-Tac-Toe via RL

- Play many games against the opponent
 - To select our moves we examine the states that would result from each of our possible moves, and look up their current values in the table
 - Most of the time we move greedily: select the move that leads to the state with the greatest value
 - Occasionally, we move randomly; these are called exploratory moves because they let us experience states that we might otherwise never see



Solving Tic-Tac-Toe via RL

- Updating values of states
 - Change the values of the states we experience during the game
 - “back up” the value of the state after each greedy move to the state before the move
 - Current value of the earlier state is updated to be closer to the value of the later state, by moving the earlier state’s value a fraction of the way toward the value of the later state
 - $V(S_t) \leftarrow V(S_t) + \alpha[V(S_{t+1}) - V(S_t)]$
 - α : a step-size parameter
 - This is called TD-learning, a model-free method



Solving Tic-Tac-Toe via RL

- Updating values of states
 - TD-learning performs well on this task
 - If the step-size parameter is reduced properly over time, then this method converges, for any fixed opponent, to the true probabilities of winning from each state given optimal play by our player
 - The moves then taken (except on exploratory moves) are in fact the optimal moves against this (imperfect) opponent
 - The method converges to an optimal policy for playing the game against this opponent
 - If the step-size parameter is not reduced all the way to zero over time, then this player also plays well against opponents that slowly change their way of playing



Observations

- Key features of RL
 - Emphasis on learning while interacting with an environment (opponent player)
 - There is a clear goal, and correct behavior requires foresight that takes into account delayed effects of one's choices



RL Beyond Tic-Tac-Toe

- RL is applicable when there is no external adversary (i.e., “game against nature”)
- RL is not restricted to episodic tasks
 - RL is applicable when behavior continues indefinitely and rewards can be received at any time
- RL is not restricted to discrete cases
- RL can be used when the state set is very large or infinite
 - Via function approximation




RL Beyond Tic-Tac-Toe

- RL can incorporate prior knowledge
- RL is applicable when part of the state is hidden
- RL is applicable even when a short-term model of the effects of actions is lacking
 - E.g., when RL agent cannot foresee how its environment would change in response to moves



Outline

- Reinforcement Learning
- Basic Examples of RL
- Elements of RL
- Extended Example: Tic-Tac-Toe
-  **Conclusion**



Conclusion

- Reinforcement learning
 - A computational approach to understanding and automating goal-directed learning and decision making
 - Learn by an agent from direct interaction with its environment, without exemplary supervision or complete models of the environment
 - Address the computational issues that arise when learning from interaction with an environment in order to achieve long-term goals
 - Uses Markov decision process to define interaction in terms of states, actions, and rewards
 - The concepts of value and value function are key to most RL methods
 - Value function is a key difference of RL compared to evolutionary methods that search directly in policy space guided by evaluations of entire policies



Questions?