Reading of theoretical aspect of reinforcement learning – Chapter 1

Zhe Rao

- The idea that we learn by interacting with our environment is probably the first to occur to us when we think about the nature of learning
    - ➢ Exercising the connection with the environment produces a wealth of information about causes and effect, and the consequences of action, and about what to do in order th achieve goals
    - ➢ Throughout our lives, such interactions are undoubtedly a major source of knowledge about our environment and ourselves.
    - ➢ Whether we are learning to drive a car or to hold a conversation, we are acutely aware of how our environment responds to what we do, and we seek to influence what happens through our behavior.
    - ➢ Learning from interaction is a foundational idea underlying nearly all theories of learning and intelligence

- In this book, we explore a computational approach to learning from interaction.
    - ➢ Rather than directly theorizing about how people or animals learn, we primarily explore idealized learning situations and evaluation the effectiveness of various learning methods.
    - ➢ We explore designs for machines that are effective in solving learning problems of scientific or economic interest, evaluating the designs through mathematical analysis or computational experiments.
    - ➢ The approach we explore, called reinforcement learning, is much more focused on goal-directed learning from interaction than are other approaches to machine learning.

1. *Reinforcement learning*

- Reinforcement learning is learning what to do, how to map situations to actions – so as to maximize a numerical reward signal.
    - ➢ The leaner is not told which actions to take, but instead must discover which actions yield the most reward by trying them.
    - ➢ In the most interesting and challenging cases, actions may affect not only the immediate reward, but also the next situation and, through that, all subsequent rewards
    - ➢ These two characteristics – **trial-and-error search** and **delayed reward** – are the two most important distinguishing features of reinforcement learning

- Reinforcement learning is simultaneously a problem, a class of solution methods that work well on that problem, and the field that studies this problem and its solution methods
    - ➢ It's essential to keep the three conceptually separate.

  - ➢ In particular, the distinction between problem and solution methods is very important in reinforcement learning; failing to make this distinction is the source of many confusions

- We formalize the problem of reinforcement learning using ideas from dynamical systems theory, specifically, as the optimal control of incompletely-known Markov decision processes
  - ➢ The basic idea is simply to capture the most important aspects of the real problem facing a learning agent interacting over time with its environment to achieve a goal
  - ➢ A learning agent must be able to sense the state of its environment to some extent and must be able to take actions that affect the state.
  - ➢ The agent also must have a goal or goals relating to the state of the environment.
  - ➢ Markov decision processes are intended to include just these three aspects – sensation, action, and goal – in their simplest possible forms without trivializing any of them

- Reinforcement learning is different from supervised learning, the kind of learning studied in most current research in the field of machine learning
  - ➢ The object of this kind of learning is for the system to extrapolate, or generalize, its responses so that it acts correctly in situations not present in the training set.
    - ✧ This is an important kind of learning, but alone it is not adequate for learning from interaction.
    - ✧ 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.
    - ✧ In uncharted territory – where one would expect learning to be most beneficial – an agent must be able to learn from its own experience.

- Reinforcement learning is also different from what machine learning researchers call unsupervised learning, which is typically about finding structure hidden in collections of unlabeled data.
  - ➢ Although one might be tempted to think of reinforcement learning as a kind of unsupervised learning because it does not rely on examples of correct behavior, reinforcement learning is trying to maximize a reward signal instead of trying to find hidden structure.
  - ➢ Uncovering structure in an agent's experience can certainly be useful in reinforcement learning, but by itself does not address the reinforcement learning problem of maximizing a reward signal.
  - ➢ We therefore consider reinforcement learning to be a third machine learning paradigm, alongside supervised learning and unsupervised learning and perhaps other paradigms

- One of the challenges that arise in reinforcement learning, and not in other kinds of learning, is the trade-off between exploration and exploitation
  - ➢ To obtain a lot of reward, a reinforcement learning agent must prefer actions that it has tried in the past and found to be effective in producing reward.
  - ➢ But to discover such actions, it has to try actions that it has not selected before.
  - ➢ The agent has to exploit what it has already experienced in order to obtain reward, but it also has to explore in order to make better action selections in the future
  - ➢ The agent must try a variety of actions and progressively favor those that appear to be best.
  - ➢ On a stochastic task, each exploration – exploitation dilemma has been intensively studies by mathematicians for many decades, yet remains unresolved

- ➢ For now, we simply note that the entire issue of balancing exploration and exploitation does not even arise in supervised and unsupervised learning.

- Another key feature of reinforcement learning is that it explicitly considers the whole problem of a goal-directed agent interacting with an uncertain environment.
  - ➢ This is in contrast to many approaches that consider subproblems without addressing how they might fit into a larger picture
  - ➢ Other researchers have developed theories of planning with general goals, but without considering planning's role in real-time decision making, or the question of where the predictive models necessary for planning world come from
- Reinforcement learning takes the opposite track, starting with a complete, interactive, goal-seeking agent.
  - ➢ All reinforcement learning agents have explicit goals, can sense aspects of their environments, and can choose actions to influence their environments.
  - ➢ Moreover, it is usually assumed from the beginning that the agent has to operate despite significant uncertainty about the environment
  - ➢ When reinforcement learning involves planning, it has to address the interplay between planning and real-time action selection, as well as the question of how environment models are required and improved.
  - ➢ When reinforcement learning involves supervised learning, it does so for specific reasons that determine which capabilities are critical and which are not.
  - ➢ For learning research to make progress, important subproblems have to be isolated and studied, but they should be subproblems that play clear roles in complete, interactive, goal-seeking agents, even if all the details of the complete agent cannot yet be filled in

- By a complete, interactive, goal-seeking agent we do not always mean something like a complete organism or robot
  - ➢ These are clearly examples, but a complete, interactive, goal-seeking agent can also be a component of a larger behaving system
  - ➢ In this case, the agent directly interacts with the rest of the larger system and indirectly interacts with the larger system's environment.
    - ✧ A simple example is an agent that monitors the charge level of robot's battery and sends commands to the robot's control architecture.
    - ✧ This agent's environment is the rest of the robot together with the robot's environment.

- One of the most exciting aspects of modern reinforcement learning is its substantive and fruitful interactions with other engineering and scientific disciplines.
  - ➢ Reinforcement learning is part of a decades-long trend within artificial intelligence and machine learning toward greater integration with statistics, optimization, and other mathematical subjects.
  - ➢ More distinctively, reinforcement learning has also interacted strongly with psychology and neuroscience, with substantial benefits going both ways.
    - ✧ Of all the forms of machine learning, reinforcement learning is the closest to the kind of learning that humans and other animals do, and many of the core algorithms of reinforcement learning were originally inspired by biological learning systems.

        - Reinforcement learning 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

2. Elements of Reinforcement Learning

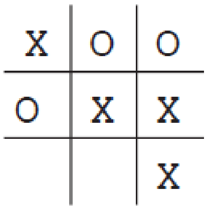
- Beyond the agent and the environment, one can identify four main sub-elements of a reinforcement learning system: a policy, a reward signal, a value function, and, optionally, a model of the environment.
    - A **policy** defines the learning agent's way of behaving at a given time.
        - Roughly speaking, a policy is a mapping from perceived states of the environment to actions to be taken when in those states.
        - In some cases the policy may be a simple function or lookup table, whereas in others it may involve extensive computation such as a search process.
        - The policy is the core of a reinforcement learning agent in the sense that it alone is sufficient to determine behavior.
        - In general, policies may be stochastic, specifying probabilities for each action.
    - A **reward signal** defines the goal of a reinforcement learning problem
        - On each time step, the environment sends to the reinforcement learning agent a single number called the reward
            - The agent's sole objective is to maximize the total reward it receives over the long run.
            - The reward signal thus defines what are the good and bad events for the agent
        - The 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 some other action in that situation in the future.
            - In general reward signals may be stochastic functions of the state of the environment and the actions taken
        - Whereas rewards determine the immediate, intrinsic desirability of environmental states, values indicate the long-term desirability of states after taking into account the state that are likely to follow and the rewards available in those states.
            - For example, 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.
        - Rewards are in a sense primary, whereas values, as predictions of rewards, are secondary
            - Without rewards there could be no values, and the only purpose of estimating values is to achieve more reward.
            - Nevertheless, it is values with which we are most concerned when making and evaluating decisions.
            - Action choices are made based on value judgements. We seek actions that bring about states of highest value, not highest reward.
        - Unfortunately, it is much harder to determine values than it is to determine rewards.
            - Rewards are basically given directly by the environment, but values must be estimated and re-estimated from the sequences of observations an agent makes over its entire lifetime

        - The central role of value estimation is arguably the most important thing that has been learned about reinforcement learning over the last six decades.
    - The final element of some reinforcement learning system is a **model of the environment**
        - This is something that mimics the behavior of the environment, or more generally, that allows inferences to be made about how the environment will behave. Those models are used for planning
        - Methods for solving reinforcement learning problems that use models and planning are called **model-based methods**, as opposed to simpler model-free methods that are explicitly trial-and-error learners – viewed as almost the opposite of planning.

3. Limitations and Scope

- The formal definition of state as we use in here is given by the framework of Markov decision process.
    - More generally, we encourage the reader to follow the informal meaning and think of the state as whatever information is available to the agent about its environment.
    - Most of the reinforcement learning methods we consider in this book are structured around estimating value functions, but it is not strictly necessary to do this to solve reinforcement learning problems
        - There are methods 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.
        - We call these evolutionary methods
        - If the space of policies is sufficiently small, or can be structured so that good policies are common or easy to find – or if a lot of time is available for the search – then evolutionary methods can be effective. In addition, evolutionary methods have advantages on problems in which the learning agent cannot sense the complete state of its environment
        - Our focus is on reinforcement learning methods that learn while interacting with the environment, which evolutionary methods do not do.

4. An extended example: Tic-Tac-Toe



X	O	O
O	X	X
		X

- For the moment, in fact, let us consider draws and losses to be equally bad for us. How might we construct a player that will find the imperfections in its opponent's play and learn to maximize its chances of winning?
    - Classical optimization methods for sequential decision problems, such as dynamic programming, can compute an optimal solution for any opponent, but require as input a complete specification of that opponent, including the probabilities with which the opponent makes each move in each board state.
    - On the other hand, such information can be estimated from experience, in this case by playing many games against the opponent.
        - About the best one can do on this problem is first to learn a model of the opponent's behavior, up to some level of confidence, and then apply dynamic programming to compute an optimal solution given the approximate opponent model.
        - In the end, this is not that different from some of the reinforcement learning methods we examine later in this book

- Here is how the tic-tac-toe problem would be approached with a method making use of a value function.
    - First we would set up a table of numbers, one for each possible state of the game.
        - Each number will be the latest estimate of the probability of our winning from that state. We treat this estimate as the state's value, and the whole table is the learned value function
        - We set the initial values of all the other states to 0.5, representing a guess that we have a 50% chance of winning.
    - We then play many games against the opponent.
        - To select our moves we examine the states that would result from each of our possible moves (one for each blank space on the board)and look up their current values in the table.
            - Most of the time we move greedily, selecting the move that leads to the state with greatest value, that is, with the highest estimated probability of winning.
            - Occasionally, however, we select randomly from among the other moves instead. There are called exploratory moves because they cause us to experience states that we might otherwise never seen.

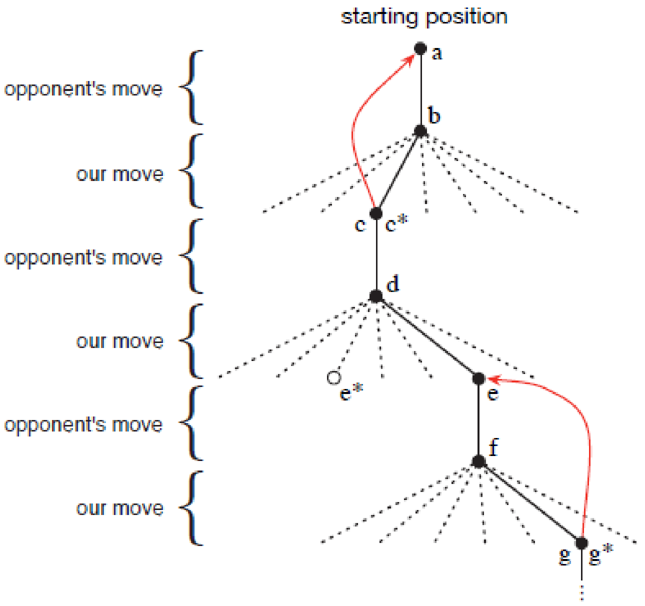


Figure 1.1: A sequence of tic-tac-toe moves. The solid black lines represent the moves taken during a game; the dashed lines represent moves that we (our reinforcement learning player) considered but did not make. The * indicates the move currently estimated to be the best. Our second move was an exploratory move, meaning that it was taken even though another sibling move, the one leading to e*, was ranked higher. Exploratory moves do not result in any learning, but each of our other moves does, causing updates as suggested by the red arrows in which estimated values are moved up the tree from later nodes to earlier nodes as detailed in the text.

- While we are playing, we change the values of the states in which we find ourselves during the game. We attempt to make them more accurate estimates of the probabilities of winning.
  - To do this, we "back up" the value of the state after each greedy move to the state before the move , as suggested by the arrows in the graph.
  - More precisely, the current value of the earlier state is updated to be closer to the value of the later state.
    - This can be done by moving the earlier state's value a fraction of the way toward the value of the later state.
- If we let S_t denote the state before the greedy move, and S_{t+1} the state after that move, then the update to the estimated value of S_t, denoted $V(S_t)$ can be written as

$$V_{new}(S_t) = V_{old}(S_t) + \alpha[V(S_{t+1}) - V(S_t)]$$

  - Where α is a small positive fraction called the step-size parameter, which influences the rate of learning.
  - This update rule is an example of a temporal-difference learning method, so called because its changes are based on the difference between estimates at two successive times

- The method described above performs quite well on this task.
  - For example, 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.
  - Furthermore, the moves then taken (except on exploratory moves) are in fact the optimal moves against this (imperfect) opponent.

- In other words, 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

- This example illustrates the differences between evolutionary methods and methods that learn value functions.
  - To evaluate a policy, an evolutionary method holds the policy fixed and plays many games against the opponent or simulated many games using a model of the opponent.
    - The frequency of wins gives an unbiased estimate of the probability of winning with that policy, and can be used to direct the next policy selection.
    - But each policy change is made only after many games, and only the final outcome of each game is used: what happens during the games is ignored
  - Value function methods, in contrast, allow individual states to be evaluated.
  - In the end, evolutionary and value function methods both search the space of policies, but learning a value function takes advantage of information available during the course of play

- Although tic-tac-toe is a two-person game, reinforcement learning also applies in the case in which there is no external adversary, that is, in the case of a “game against nature”.
  - Reinforcement learning also is not restricted to problems in which behavior breaks down into separate episodes, like the separate games of tic-tac-toe, with reward only at the end of each episode.
  - It is just as applicable when behavior continues indefinitely and when rewards of various magnitudes can be received at any time.
  - Reinforcement learning is also applicable to problems that do not even break down into discrete time steps like the example.
    - The general principles apply to continuous-time problems as well, although the theory gets more complicated and we omit it from this introductory treatment.

- Tic-tac-toe has a relatively small, finite state set, whereas reinforcement learning can be used when the state set is very large, or even infinite.
  - Consider combining the algorithm described above with an artificial neural network to learn to play backgammon.
  - With this many states, it is impossible ever to experience more than a small fraction of them
  - The artificial neural network provides the program with the ability to generalize from its experience, so that in new states it selects moves based on information saved from similar states faced in the past, as determined by the network
- How well a reinforcement learning system can work in problems with such large state sets is intimately tied to how appropriately it can generalize from past experience.
  - It is in this role that we have the greatest need for supervised learning methods within reinforcement learning
  - In the tic-tac-toe example, learning started with no prior knowledge beyond the rules of the game
    - Prior information can be incorporated into reinforcement learning in a variety of ways that can be critical for efficient learning

  - ➢ We also have access to the true state in the tic-tac-toe example, whereas reinforcement learning can also be applied when part of the state is hidden, or when different states appear to the learner to be the same
  - ➢ Finally, the tic-tac-toe player was able to look ahead and know the states that would result from each of its possible moves.
    - ✧ To do this, it had to have a model of the game that allowed it to foresee how its environment would change in response to moves that it might never make.
    - ✧ Many problems are like this, but in others even a short-term model of the effects of actions is lacking.
    - ✧ Reinforcement learning can be applied in either case

- On the other hand, there are reinforcement learning methods that do not need any kind of environment model at all.
  - ➢ Model-free systems cannot even think about how their environments will change in response to a single action.
  - ➢ The tic-tac-toe player is model-free in this sense with respect to its opponent: it has no model of its opponent of any kind.
  - ➢ Because models have to be reasonably accurate to be useful, model-free methods can have advantages over more complex methods when the real bottleneck in solving a problem is the difficulty of constructing a sufficiently accurate environment model.
  - ➢ Model-free methods are also important building blocks for model-based methods.

- Reinforcement learning can be used at both high and low level in a system
  - ➢ Although the tic-tac-toe player learned only about the basic moves of the game, nothing prevents reinforcement learning from working at higher levels where each of the “actions” may itself be the application of a possibly elaborate problem-solving method.
  - ➢ In hierarchical learning systems, a reinforcement learning can work simultaneously on several levels