ECE 1508: Reinforcement Learning

Chapter 1: Introduction

Ali Bereyhi

ali.bereyhi@utoronto.ca

Department of Electrical and Computer Engineering

University of Toronto

Fall 2025

Playing in RL Framework: *Policy*

As environment gets to state S_t, agent decides for an action

↳ *We can describe this behavior via policy function*

Policy

Policy $\pi(a|s)$ *is a probability distribution over* a *conditional to* s

$$\pi(a|s) = \Pr\{A_t = a | S_t = s\}$$

+ *Why a probability distribution? Does agent always act randomly?!*
– It can either act randomly or deterministically, and *both these cases can be presented by this notation*

Playing in RL Framework: *Probabilistic Policy*

Assume we have set of possible actions

$$\mathbb{A} = \left\{a^1, \dots, a^M\right\}$$

Agents might behave probabilistic, e.g.,

$$\pi\left(a|s\right) = \begin{cases} 0.8 & a = a^1 \\ 0.2 & a = a^2 \\ 0 & \textit{otherwise} \end{cases}$$

This means that in state s

- *in 80% of cases it acts a^1*
- *in 20% of cases it acts a^2*
- *it never does any other action*

Playing in RL Framework: *Deterministic Policy*

Assume we have set of possible actions

$$\mathbb{A} = \left\{a^1, \dots, a^M\right\}$$

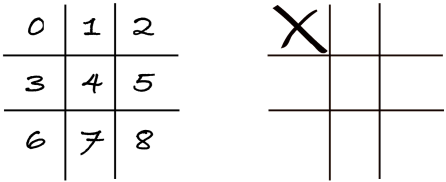
Agents might behave deterministically, e.g.,

$$\pi\left(a|s\right) = \begin{cases} 1 & a = a^1 \\ 0 & \textit{otherwise} \end{cases}$$

This means that if the agent observes state s

- *it always acts a^1*
- *it never does any other action*

Example: *Tic-Tac-Toe*



Let's look at the Tic-Tac-Toe example: assume the oponent starts at cell 0

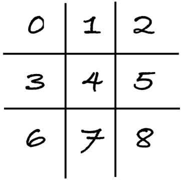
- *State is then* $S_1 = \{0\}$*: we can set the policy to*
  - ↳ *In the policy, we should have specified*

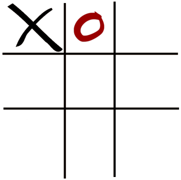
$$\pi(a|\{0\}) = \Pr\{A_1 = a|S_1 = \{0\}\} = \begin{cases} p_a & a = 1, \dots, 8 \\ 0 & a = 0 \end{cases}$$

for some p_a*'s that satisfy*

$$\sum_{a=1}^{8} p_a = 1$$

Example: *Tic-Tac-Toe*



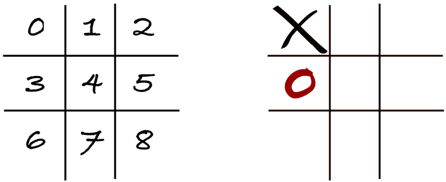


We may play deterministically: we are sure what to do after seeing opponent playing zero

↳ *We may always play* $A_1 = 1$

$$\pi\left(a \middle| \{0\}\right) = \begin{cases} 1 & a = 1 \\ 0 & a \neq 1 \end{cases}$$

Example: *Tic-Tac-Toe*

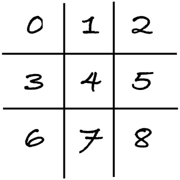


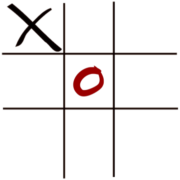
We may play deterministically: we are sure what to do after seeing opponent playing zero

↳ *We may always play* $A_1 = 3$

$$\pi\left(a|\left\{0\right\}\right) = \begin{cases} 1 & a = 3 \\ 0 & a \neq 3 \end{cases}$$

Example: *Tic-Tac-Toe*



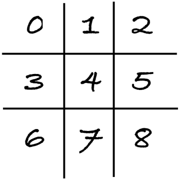


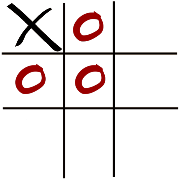
We may play deterministically: we are sure what to do after seeing opponent playing zero

↳ *We may always play* $A_1 = 4$

$$\pi\left(a|\left\{0\right\}\right) = \begin{cases} 1 & a = 4 \\ 0 & a \neq 4 \end{cases}$$

Example: *Tic-Tac-Toe*



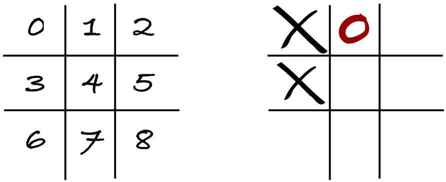


We may play probabilistic: we behave somehow randomly after seeing opponent playing zero

↳ *We may play any next cell* $A_1 \in \{1, 3, 4\}$

$$\pi\left(a | \{0\}\right) = \begin{cases} 1/3 & a \in \{1, 3, 4\} \\ 0 & a \notin \{1, 3, 4\} \end{cases}$$

Example: *Tic-Tac-Toe*

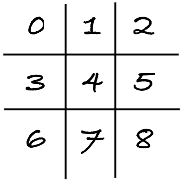


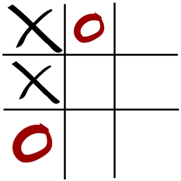
Say we acted $A_1 = 1$, and the opponent plays 3 after our move: policy should tell us the next move

↳ *State changes to $S_2 = \{0, 1, 3\}$ and our policy should say*

$$\pi \left(a | \left\{ 0, 1, 3 \right\} \right) = \Pr \left\{ A_2 = a | S_2 = \left\{ 0, 1, 3 \right\} \right\}$$

Example: *Tic-Tac-Toe*





Say we acted $A_1 = 1$, and the opponent plays 3 after our move: policy should tell us the next move

↳ *For instance, we could play deterministically as*

$$\pi\left(a | \{0,1,3\}\right) = \begin{cases} 1 & a = 6 \\ 0 & a \neq 6 \end{cases}$$

i.e., we always circle cell 6 in this state of the board

Playing in RL Framework: *Value Function*

As agent gets into a new state, it can use its policy to *estimate future return*

↳ *We call this estimate the value of the state*

Value Function

Assume that agent acts with policy $\pi(a|s)$ *at state* s*; then, the value of this state is defined as*

$$v_\pi(s) = \mathbb{E}\left\{G_t | S_t = s\right\}$$

Because it depends on policy π

+ *Why do we have index* π*?*
– Because this value depends on the policy!

Playing in RL Framework: *Value Function*

Let's break it down: *we first recall that*

Future return at time t

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots$$

for some discount factor $0 \leqslant \gamma \leqslant 1$

Also, we recall that

Rewarding function is going to determine the next reward

$$R_{t+1} = \mathcal{R}\left(S_t, A_t\right)$$

and transition function determines the next state

$$S_{t+1} = \mathcal{P}\left(S_t, A_t\right)$$

Playing in RL Framework: *Value Function*

Value of state s is

$$v_\pi(s) = \mathbb{E}\{G_t | S_t = s\} = \mathbb{E}\left\{R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s\right\}$$

Say we are at time t: *at this time we see state $S_t = s$*

- *Policy $\pi(a|s)$ tells us what to play next*
  - ↳ *For instance, it deterministically specifies A_t*
- *Rewarding function specifies the reward of next time step*

$$R_{t+1} = \mathcal{R}(S_t, A_t)$$

- *Transision function specifies the next state*

$$S_{t+1} = \mathcal{P}(S_t, A_t)$$

↳ *Both rewarding and transition could be probabilistic*

Playing in RL Framework: *Value Function*

> *Value of state s is*
>
> $$v_\pi(s) = \mathbb{E}\{G_t | S_t = s\} = \mathbb{E}\left\{R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s\right\}$$

Given rewarding function and policy: *distribution of R_{t+1} is found*

- *We compute the probability of each possible rewards, i.e.,*

$$p_\ell(s, a) = \Pr\left\{R_{t+1} = r^\ell | S_t = s, A_t = a\right\}$$

for $\ell = 1, \dots, L$ and each possible action a

↳ *We can hence compute the first term*

$$\mathbb{E}_\pi\{R_{t+1} | S_t = s\} = \sum_{\ell=1}^{L} \sum_{m=1}^{M} p_\ell(s, a^m)\, \pi(a^m | s)\, r^\ell$$

Playing in RL Framework: *Value Function*

Note that *distribution of* R_{t+1} *can change, if we change policy*

Example: *Say rewarding function says* $\mathcal{R}\left(s, a^1\right) = 0$ *and* $\mathcal{R}\left(s, a^2\right) = 1$; *now, consider the following policies*

$$\pi^1\left(a|s\right) = \begin{cases} 1 & a = a^1 \\ 0 & a \neq a^1 \end{cases} \qquad \pi^2\left(a|s\right) = \begin{cases} 1 & a = a^2 \\ 0 & a \neq a^2 \end{cases}$$

Therefore, we could say

$$\mathbb{E}_{\pi^1}\left\{R_{t+1}|S_t = s\right\} = 0 \qquad \mathbb{E}_{\pi^2}\left\{R_{t+1}|S_t = s\right\} = 1$$

Playing in RL Framework: *Value Function*

Value of state s is

$$v_\pi(s) = \mathbb{E}\{G_t|S_t = s\} = \mathbb{E}\left\{R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots |S_t = s\right\}$$

For time $t+1$: *we have specified state S_{t+1} via transition function*

- *Policy $\pi(a|S_{t+1})$ tells us what to play next, i.e., A_{t+1}*
  - ↳ *Note that S_{t+1} is in general a random variable*
- *Rewarding function specifies the reward of next time step*

$$R_{t+2} = \mathcal{R}(S_{t+1}, A_{t+1})$$

↳ *We can hence compute the second term*

$$\mathbb{E}_\pi\{R_{t+2}|S_t = s\}$$ ⟵ Expectation over S_{t+1}, A_{t+1} and A_t

Playing in RL Framework: *Value Function*

> *Value of state s is*
>
> $$v_\pi(s) = \mathbb{E}\{G_t | S_t = s\} = \mathbb{E}\left\{R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots | S_t = s\right\}$$

So if we set a specific policy π, we could keep on

- $\mathbb{E}_\pi\{R_{t+3} | S_t = s\}$

  $\ldots$

and we could estimate future return for a given policyπ as

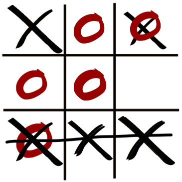
$$v_\pi(s) = \mathbb{E}_\pi\{R_{t+1} | S_t = s\} + \gamma \mathbb{E}_\pi\{R_{t+2} | S_t = s\} + \gamma^2 \mathbb{E}_\pi\{R_{t+3} | S_t = s\} + \cdots$$

Value Example: *Tic-Tac-Toe*

Assume we play Tic-Tac-Tao with a pretty deterministic opponent:

We know that opponent takes the following strategy:

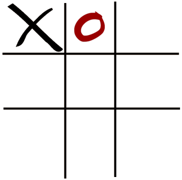
- *if corners are available without chance of loosing, it plays a corner*
  - ↳ *it starts with cross-corner*
  - ↳ *it plays other corners afterwards*
- *if a line can be filled by us in next move, it blocks the line*
  - ↳ *if multiple lines are available, it blocks one of them at random*
- *if there is a line to be filled, it fills and wins*



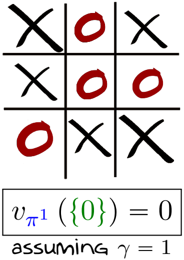
Value Example: *Tic-Tac-Toe – Policy I*

We now take the following policy

- *if there is no chance of loosing, we play next cell*
  - ↳ *if next cell in the row available, we play that cell*
  - ↳ *if next cell in the row not available, we play cell below*
  - ↳ *if neither is available, we play some cell at random*
- *if there is a line possibly filled in next move by opponent, we block it*
  - ↳ *if multiple lines are available, we block one of them at random*
- *if there is a line to be filled by us, we fill and win*



Value Example: *Tic-Tac-Toe – Policy I*



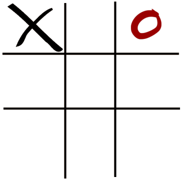
Say the game starts at state $s = \{0\}$ with policy I: π^1

- *Our next move will be* $A_0 = 1$
- *Opponent will play* $O_1 = 8$
  - ↳ *this leads to new state* $S_1 = \{0, 1, 8\}$
- *Our next move will be* $A_1 = 4$
- *Opponent will play* $O_2 = 7$
  - ↳ *this leads to new state* $S_2 = \{0, 1, 4, 7, 8\}$
- *Our next move will be* $A_2 = 6$
- *Opponent will play* $O_3 = 2$
  - ↳ *this leads to new state* $S_3 = \{0, 1, 2, 4, 6, 7, 8\}$
- *Our next move will be* $A_3 = 5$
- *Opponent will play* $O_4 = 3$
  - ↳ *this leads to new state* $S_4 = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$

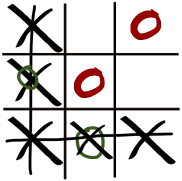
Value Example: *Tic-Tac-Toe – Policy II*

Let's now change the policy

- *if there is no chance of loosing, we play a corner*
  - ↳ *if a corner is available in the row, we play that corner*
  - ↳ *if no corner is available in the row, we play corner in the same column*
  - ↳ *if neither is available, we play some random cell*
- *if there is a line possibly be filled in next round by opponent, we block it*
  - ↳ *if multiple lines are available, we block one of them at random*
- *if there is a line to be filled by us, we fill and win*



Value Example: *Tic-Tac-Toe – Policy II*



$v_{\pi^2}(\{0\}) = -1$

assuming $\gamma = 1$

Say the game starts at state $s = \{0\}$ with policy I: π^2

- *Our next move will be $A_0 = 2$*
- *Opponent will play $O_1 = 8$*
  - ↳ *this leads to new state $S_1 = \{0, 2, 8\}$*
- *Our next move will be $A_1 = 4$*
- *Opponent will play $O_2 = 6$*
  - ↳ *this leads to new state $S_2 = \{0, 2, 4, 6, 8\}$*
- *Our next move will be random*
  - ↳ *with 50% chance we play $A_2 = 7$*
    - ↳ *Opponent will play $O_3 = 3$*
  - ↳ *with 50% chance we play $A_2 = 3$*
    - ↳ *Opponent will play $O_3 = 7$*
- *With any possible move, we end up losing*

Playing in RL Framework: *Value Function*

Value function can be also calculate for each possible next action

↳ *We call this function the action-value of the state-action pair*

Action-Value Function

Assume that agent acts with policy $\pi(a|s)$ at state s; then, the action-value of this state is defined as

$$q_\pi(s, a) = \mathbb{E}_\pi\{G_t | S_t = s, A_t = a\}$$

Because it depends on policy π

Playing in RL Framework: *Action-Value Function*

> *Action-Value of state-action pair (s, a) is*
>
> $$q_\pi(s,a) = \mathbb{E}_\pi\left\{R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s, A_t = a\right\}$$

Given rewarding function: *distribution of R_{t+1} for given pair (s, a) is*

- *We compute the probability of each possible rewards, i.e.,*

$$p_\ell(s,a) = \Pr\left\{R_{t+1} = r^\ell | S_t = s, A_t = a\right\}$$

for $\ell = 1, \dots, L$ and each possible action a

↳ *We can hence compute the first term*

$$\mathbb{E}_\pi\left\{R_{t+1} | S_t = s, A_t = a\right\} = \sum_{\ell=1}^{L} p_\ell(s,a)\, r^\ell$$

Playing in RL Framework: *Action-Value Function*

Action-Value of state-action pair (s, a) *is*

$$q_\pi(s,a) = \mathbb{E}_\pi\left\{R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s, A_t = a\right\}$$

For time $t+1$: *we have specified state* S_{t+1} *via transition function*

- *Policy* $\pi(a|S_{t+1})$ *tells us what to play next, i.e.,* A_{t+1}
  - ↳ *Note that* S_{t+1} *is in general a random variable*
- *Rewarding function specifies the reward of next time step*

$$R_{t+2} = \mathcal{R}(S_{t+1}, A_{t+1})$$

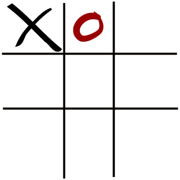
↳ *We can hence compute the second term*

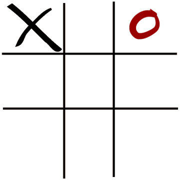
$$\mathbb{E}_\pi\left\{R_{t+2} | S_t = s, A_t = a\right\}$$ ⇜ Expectation over S_{t+1} and A_{t+1}

Action-Value Example: *Tic-Tac-Toe – Hybrid Policy*

We play Tic-Tac-Tao with the same opponent: this time however we take a hybrid approach

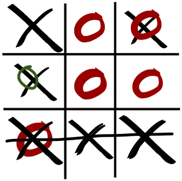
- *with 50% chance we stick to policy I*
- *with 50% chance we stick to policy II*





Let's denote this hybrid policy by $\bar{\pi}$

Action-Value Example: *Tic-Tac-Toe – Hybrid Policy*



Say the game starts at state $s = \{0\}$

- *If we decide to stick to policy I*
  - ↳ *next move will be $A_0 = 1$*
  - ↳ *we end up drawing*

$$q_{\bar{\pi}}(\{0\}, 1) = 0$$

- *If we decide to stick to policy II*
  - ↳ *next move will be $A_0 = 2$*
  - ↳ *we end up losing*

$$q_{\bar{\pi}}(\{0\}, 2) = -1$$

We see it in the assignment that with this hybrid approach we have

$$v_{\bar{\pi}}(\{0\}) = 0.5 q_{\bar{\pi}}(\{0\}, 1) + 0.5 q_{\bar{\pi}}(\{0\}, 2) = -0.5$$

Playing in RL Framework: *Optimal Policy*

+ *What is the application of value functions?*
– They let us compare policies

We can compare π^1 to π^2 using the value function

Better Policy

We say policy π^1 is better than policy π^2, and denote it by $\pi^1 \geqslant \pi^2$ if

$$v_{\pi^1}(s) \geqslant v_{\pi^2}(s)$$

for all $s \in \mathbb{S}$

Playing in RL Framework: *Optimal Policy*

It is important to note that when we say $\pi^1 \geqslant \pi^2$; it means that by playing π^1, we expect higher or equal future return than π^2, no matter at which state we start using policy π^1

This definition helps us defining optimal policy: *simply speaking*

optimal policy is the policy that is the best

Optimal Policy

Policy $\pi^\star$ is called optimal policy if for any possible policy π, we have

$$\pi^\star \geqslant \pi$$

Playing in RL Framework: *Optimal Policy*

Optimal Policy: *Alternative Definition*

Policy $\pi^\star$ is called optimal policy if it maximizes the value for each state

$$\pi^\star = \operatorname*{argmax}_{\pi} v_\pi(s)$$

for all $s \in \mathbb{S}$

+ *Can we guarantee that the optimal policy always exists?*
– Yes!
+ *Why didn't we define it based on action-value function?*
– Well! We could, but we end up with the same result!

Playing in RL Framework: *Optimal Policy*

Basic Property of Optimal Policy

Let $\pi^\star$ be the optimal policy; then,

$$v_\star(s) = v_{\pi^\star}(s) = \max_\pi v_\pi(s)$$

for all $s \in \mathbb{S}$, and

$$q_\star(s,a) = q_{\pi^\star}(s,a) = \max_\pi q_\pi(s,a)$$

for all $s \in \mathbb{S}$ and $a \in \mathbb{A}$

Attention

Optimal policy is not necessarily unique: we may have multiple optimal policies

Ultimate Goal in RL: *Finding Optimal Policy*

Ultimate goal in an RL problem is to find the optimal policy

But there are a bunch of challenges in this way

1. *How can we get a model of environment?*
   - ↳ *Maybe, we could assume some model*
     - ↳ *We assume some transition and rewarding functions*
     - ↳ *Our RL framework is based on a model*
   - ↳ *Maybe, we could estimate what we need directly from enough samples*
     - ↳ *We directly compute value or policy from samples, e.g.,*

$$q_\pi(s, a) = \mathbb{E}\{G_t | S_t = s, A_t = a\} \approx \text{average from samples}$$

     - ↳ *Our RL framework is free of a model*

High Level Classification of RL Methods

Model-based RL

- Bellman Equation
  - value iteration
  - policy iteration

Model-free RL

- on-policy methods
  - temporal difference
  - Monte Carlo
  - SARSA
- off-policy methods
  - Q-learning

Ultimate Goal in RL: *Finding Optimal Policy*

Ultimate goal in an RL problem is to find the optimal policy

But there are a bunch of challenges in this way

2. *How can we apply either approach in environments with huge # of states?*
   - ↳ *Maybe, we could use neural networks*
     - ↳ *We approximate the target function via a neural network, e.g.,*

$$v_{\star}\left(s\right) \approx \mathrm{CNN}\left(s|\mathbf{w}\right)$$

     - ↳ *We collect some samples and train the network based on them*
     - ↳ *We use that approximation instead of true function*

High Level Classification of RL Methods