Reinforcement learning

- We want to learn a control policy: $\pi : X \rightarrow A$
- We see examples of $x$ (but outputs $a$ are not given)
- Instead of $a$ we get a feedback $r$ (reinforcement, reward) from a critic quantifying how good the selected output was

- The reinforcements may not be deterministic
- **Goal:** find $\pi : X \rightarrow A$ with the best expected reinforcements
Gambling example

- **Game**: 3 different biased coins are tossed
  - The coin to be tossed is selected randomly from the three options and I always see which coin I am going to play next
  - I make bets on head or tail and I always wage $1
  - If I win I get $1, otherwise I lose my bet

- **RL model**:
  - **Input**: $X$ – a coin chosen for the next toss,
  - **Action**: $A$ – choice of head or tail,
  - **Reinforcements**: $\{1,-1\}$

- **A policy** $\pi : X \rightarrow A$

  **Example**:
  
  $\pi : \begin{array}{c}
  \text{Coin1$\rightarrow$head} \\
  \text{Coin2$\rightarrow$tail} \\
  \text{Coin3$\rightarrow$head}
  \end{array}$

- **Learning goal**: find $\pi : X \rightarrow A$ such that $\pi : \begin{array}{c}
  \text{Coin1$\rightarrow$?} \\
  \text{Coin2$\rightarrow$?} \\
  \text{Coin3$\rightarrow$?}
  \end{array}$

  maximizing future expected profits

  $E(\sum_{t=0}^{\infty} \gamma^t r_t) \quad 0 \leq \gamma < 1$

  a discount factor = present value of money
### Expected rewards

- Expected rewards for $\pi : X \rightarrow A$

$$
\begin{align*}
E(\sum_{t=0}^{\infty} \gamma^t r_t) & \quad \text{Expectation over many possible discounted} \\
& \quad \text{reward trajectories for } \pi : X \rightarrow A
\end{align*}
$$

### Agent navigation example

- **Agent navigation in the Maze:**
  - 4 moves in compass directions
  - Effects of moves are stochastic – we may wind up in other than intended location with a non-zero probability
  - **Objective:** learn how to reach the goal state in the shortest expected time
Agent navigation example

- **Input:** X – position of an agent
- **Output:** A – a move
- **Reinforcements:** R
  - -1 for each move
  - +100 for reaching the goal
- **A policy:** \( \pi : X \rightarrow A \)

<table>
<thead>
<tr>
<th>Position</th>
<th>Move</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>right</td>
</tr>
<tr>
<td>2</td>
<td>right</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>left</td>
</tr>
</tbody>
</table>

**Goal:** find the policy maximizing future expected rewards

\[
E\left( \sum_{t=0}^{\infty} \gamma^t r_t \right) \quad 0 \leq \gamma < 1
\]

Objectives of RL learning

- **Objective:**
  - Find a mapping \( \pi^* : X \rightarrow A \)
    That maximizes some combination of future reinforcements (rewards) received over time
- **Valuation models** (quantify how good the mapping is):
  - Finite horizon model
    \[
    E\left( \sum_{t=0}^{T} r_t \right) \quad \text{Time horizon: } T > 0
    \]
  - Infinite horizon discounted model
    \[
    E\left( \sum_{t=0}^{\infty} \gamma^t r_t \right) \quad \text{Discount factor: } 0 \leq \gamma < 1
    \]
  - Average reward
    \[
    \lim_{T \rightarrow \infty} \frac{1}{T} E\left( \sum_{t=0}^{T} r_t \right)
    \]
Exploration vs. Exploitation

- The (learner) actively interacts with the environment:
  - At the beginning the learner does not know anything about the environment
  - It gradually gains the experience and learns how to react to the environment
- **Dilemma (exploration-exploitation):**
  - After some number of steps, should I select the best current choice (exploitation) or try to learn more about the environment (exploration)?
  - **Exploitation** may involve the selection of a sub-optimal action and prevent the learning of the optimal choice
  - **Exploration** may spend too much time on trying bad currently suboptimal actions

Effects of actions on the environment

**Effect of actions on the environment** (next input $x$ to be seen)

- No effect, the distribution over possible $x$ is fixed; action consequences (rewards) are seen immediately,
- Otherwise, distribution of $x$ can change; the rewards related to the action can be seen with some delay.

Leads to two forms of **reinforcement learning:**

- **Learning with immediate rewards**
  - Gambling example
- **Learning with delayed rewards**
  - Agent navigation example: move choices affect the state of the environment (position changes), a big reward at the goal state is delayed
**RL with immediate rewards**

- **Game:** 3 different biased coins are tossed
  - The coin to be tossed is selected randomly from the three options and I always see which coin I am going to play next
  - I make bets on head or tail and I always wage $1
  - If I win I get $1, otherwise I lose my bet

- **RL model:**
  - **Input:** $X$ – a coin chosen for the next toss
  - **Action:** $A$ – head or tail bet
  - **Reinforcements:** \{1, -1\}

- **Learning goal:** find $\pi : X \rightarrow A$

  maximizing the future expected profits over time

  $$E(\sum_{t=0}^{\infty} \gamma^t r_t) \quad 0 \leq \gamma < 1$$

  a discount factor

---

**RL with immediate rewards**

- **Expected reward**

  $$E(\sum_{t=0}^{\infty} \gamma^t r_t) \quad 0 \leq \gamma < 1$$

- **Immediate reward case:**
  - Reward for the choice becomes available immediately
  - Our action does not affect the environment and thus future rewards

  $$E(\sum_{t=0}^{\infty} \gamma^t r_t) = E(r_0) + E(\gamma r_1) + E(\gamma^2 r_2) + \cdots$$

  $r_0, r_1, r_2 \ldots$ Rewards for every step

  - Expected one step reward for input $x$ and the choice $a$

  $$R(x, a)$$
**RL with immediate rewards**

**Immediate reward case:**
- Reward for the choice \( a \) becomes available immediately
- **Expected reward for the input** \( x \) **and choice** \( a \): \( R(x, a) \)
  - For the gambling problem it is:
    \[
    R(x, a_i) = \sum_j r(\omega_j | a_i, x) P(\omega_j | x, a_i)
    \]
  - \( \omega_j \) - a future outcome of the coin toss
  - Recall the definition of the expected loss
- **Expected one step reward for a strategy** \( \pi : X \to A \)
  \[
  R(\pi) = \sum_x R(x, \pi(x)) P(x)
  \]
  \( R(\pi) \) is the expected reward for \( r_0, r_1, r_2 \ldots \)

---

**RL with immediate rewards**

- **Expected reward**
  \[
  E(\sum_{i=0}^{\infty} \gamma^i r_i) = E(r_0) + E(\gamma r_1) + E(\gamma^2 r_2) + \ldots
  \]
- **Optimizing the expected reward**:
  \[
  \max_\pi E(\sum_{i=0}^{\infty} \gamma^i r_i) = \max_\pi \sum_{i=0}^{\infty} \gamma^i E(r_i) = \max_\pi \sum_{i=0}^{\infty} \gamma^i R(\pi) = \max_\pi R(\pi) \sum_{i=0}^{\infty} \gamma^i
  \]
  \[
  = (\sum_{i=0}^{\infty} \gamma^i) \max_\pi R(\pi)
  \]
  \[
  \max R(\pi) = \max_\pi \sum_x R(x, \pi(x)) P(x) = \sum_x P(x) [\max_\pi R(x, \pi(x))]
  \]
  **Optimal strategy**: \( \pi^* : X \to A \)
  \[
  \pi^*(x) = \arg \max_a R(x, a)
  \]
RL with immediate rewards

- **We know that** $\pi^*(x) = \arg \max_a R(x, a)$
- **Problem**: In the RL framework we do not know $R(x, a)$
  - The expected reward for performing action $a$ at input $x$
- **How to get** $R(x, a)$?

### Solution:
- For each input $x$ try different actions $a$
- Estimate $R(x, a)$ using the average of observed rewards
  $$\tilde{R}(x, a) = \frac{1}{N_{x,a}} \sum_{i=1}^{N_{x,a}} r_{i}^{x,a}$$
- Action choice $\pi(x) = \arg \max_a \tilde{R}(x, a)$
- Accuracy of the estimate: statistics (Hoeffding’s bound)
  $$P\left( \left| \tilde{R}(x, a) - R(x, a) \right| \geq \varepsilon \right) \leq \exp \left[ -\frac{2\varepsilon^2 N_{x,a}}{(r_{\max} - r_{\min})^2} \right] \leq \delta$$
- Number of samples:
  $$N_{x,a} \geq \frac{(r_{\max} - r_{\min})^2}{2\varepsilon^2} \ln \frac{1}{\delta}$$
**RL with immediate rewards**

- **On-line (stochastic approximation)**
  - An alternative way to estimate $R(x, a)$
- **Idea:**
  - choose action $a$ for input $x$ and observe a reward $r^{x,a}$
  - Update an estimate

\[
\tilde{R}(x, a) \leftarrow (1 - \alpha)\tilde{R}(x, a) + \alpha r^{x,a} \quad \alpha \text{ - a learning rate}
\]

- **Convergence property:** The approximation converges in the limit for an appropriate learning rate schedule.
- **Assume:** $\alpha(n(x, a))$ - is a learning rate for $n$th trial of $(x, a)$ pair
- **Then the converge is assured if:**

1. \[\sum_{i=1}^{\infty} \alpha(i) = \infty\]
2. \[\sum_{i=1}^{\infty} \alpha(i)^2 < \infty\]

---

**Exploration vs. Exploitation**

- In the RL framework
  - the (learner) actively interacts with the environment.
  - At any point in time it has an estimate of $\tilde{R}(x, a)$ for any input action pair
- **Dilemma:**
  - Should the learner use the current best choice of action (exploitation)

\[
\hat{a}(x) = \arg \max_{a \in A} \tilde{R}(x, a)
\]

  - Or choose other action $a$ and further improve its estimate (exploration)
- **Different exploration/exploitation strategies exist**
Exploration vs. Exploitation

- **Uniform exploration**: Exploration parameter \( 0 \leq \varepsilon \leq 1 \)
  - Choose the “current” best choice with probability \( 1 - \varepsilon \)
    \[
    \hat{a}(x) = \arg \max_{a \in A} \hat{R}(x, a)
    \]
  - All other choices are selected with a uniform probability \( \frac{\varepsilon}{|A| - 1} \)

- **Boltzmann exploration**
  - The action is chosen randomly but proportionally to its current expected reward estimate
    \[
    p(a \mid x) = \frac{\exp\left[\frac{\hat{R}(x, a)}{T}\right]}{\sum_{a' \in A} \exp\left[\frac{\hat{R}(x, a')}{T}\right]}
    \]
  - \( T \) – is temperature parameter. **What does it do?**