REINFORCEMENT LEARNING

FUNDAMENTALS + APPLICATIONS

WEEK 2

MARKOV DECISION PROCESSES

THE MARKOV PROPERTY

MARKOV DECISION PROCESSES

A state S_t is Markov IF AND ONLY IF:

$$\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, S_2, \ldots, S_t]$$

The Markov property means a process or state is *memoryless*: what happens next *only* depends on where you are right now.

MARKOV CHAIN

MARKOV DECISION PROCESSES

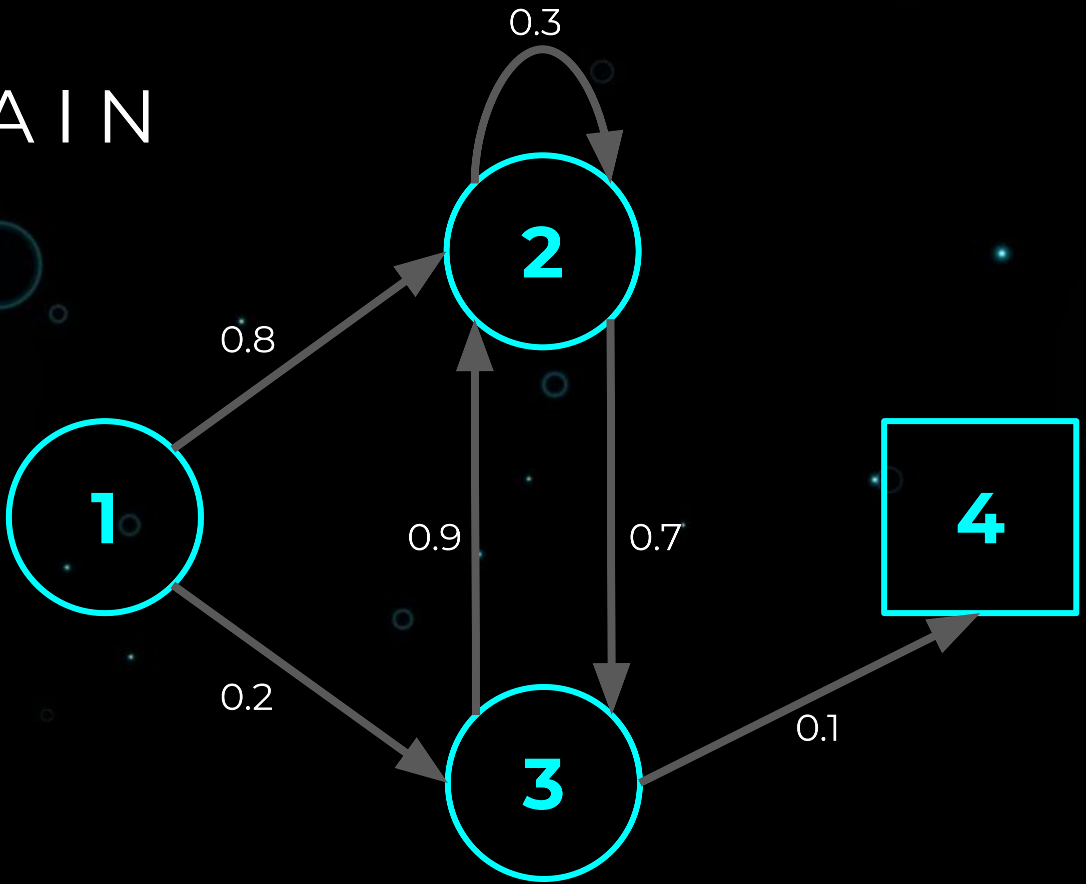
A Markov Chain | Markov Process is a tuple $(\mathcal{S}, \mathcal{P})$ where:

- $\mathcal{S}$ is a finite set of states.
- $\mathcal{P}$ is our state-transition probability matrix.

$$\mathbb{P}[S_{t+1} = s' \mid S_t = s]$$

This is a memoryless, random process.

MARKOV CHAIN



MARKOV CHAIN

STATE TRANSITION PROBABILITY MATRIX

To This State

From This State

	1	2	3	4
1	0	0.8	0.2	0
2	0	0.3	0.7	0
3	0	0.9	0	0.1
4	0	0	0	1

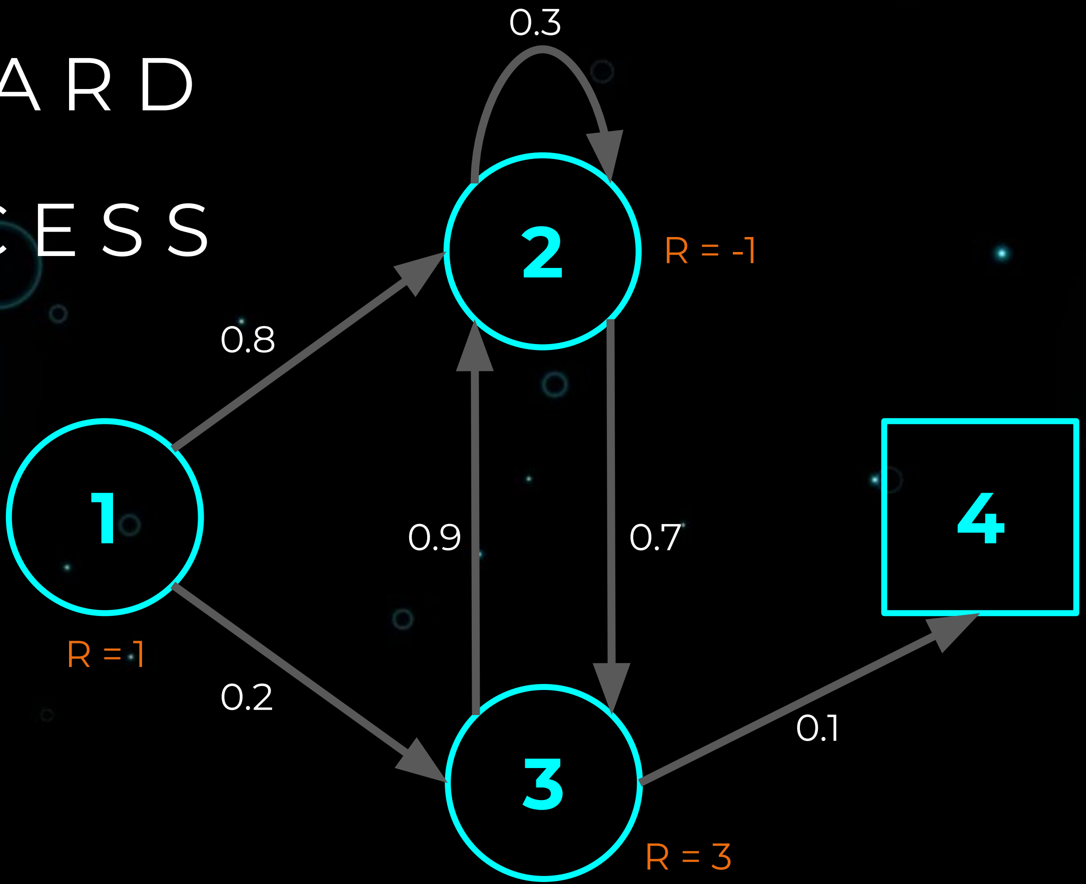
MARKOV REWARD PROCESS

MARKOV DECISION PROCESSES

A Markov Reward Process is a tuple $(\mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma)$ where:

- $\mathcal{S}$ is a finite set of states.
- $\mathcal{P}$ is our state-transition probability matrix.
- $\mathcal{R}$ is a reward function $\mathcal{R}_s = \mathbb{E}[R_{t+1} \mid S_t = s]$
- γ is a discount factor $\gamma \in [0,1]$

MARKOV REWARD PROCESS



MARKOV DECSION PROCESSES

WE FINALLY ARRIVE

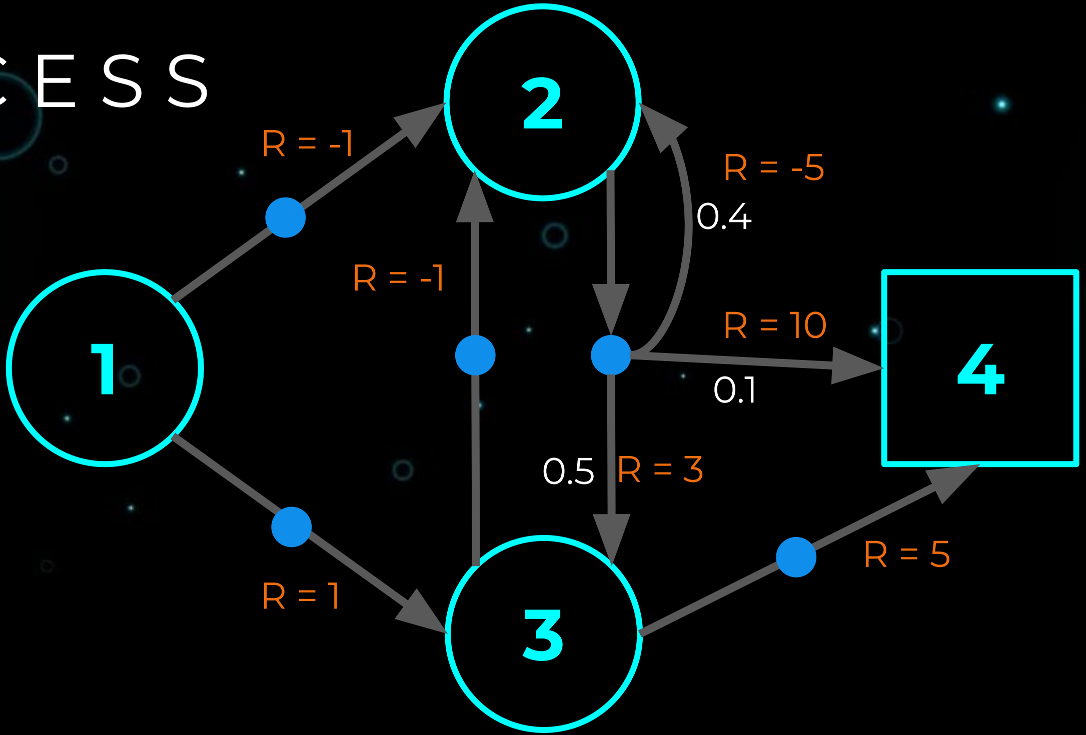
A Markov Decision Process is a tuple ($\mathscr{S}$, $\mathscr{P}$, $\mathcal{R}$, γ, $\mathscr{A}$) where:

- $\mathscr{S}$ is a finite set of states.
- $\mathscr{A}$ is a finite set of actions.
- $\mathscr{P}$ is our state-transition probability matrix.

$$\mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$$

- $\mathcal{R}$ is a reward function $\mathcal{R}_s = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$
- γ is a discount factor $\gamma \in [0,1]$

MARKOV DECISION PROCESS



RETURN

MARKOV DECISION PROCESSES

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

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

The total discounted FUTURE reward from timestep t *onward*.

STATE-VALUE FUNCTION

MARKOV DECISION PROCESSES

$$v(s) = \mathbb{E}[G_t \mid S_t = s]$$

The expected return if you start in state s.

The *long-term* value of state s.

BELLMAN EQUATIONS

MARKOV DECISION PROCESSES

$$v(s) = \mathbb{E}[G_t \mid S_t = s]$$

$$= \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \mid S_t = s]$$

$$= \mathbb{E}[R_{t+1} + \gamma (R_{t+2} + \gamma R_{t+3} + \dots) \mid S_t = s]$$

$$= \mathbb{E}[R_{t+1} + \gamma G_{t+1} \mid S_t = s]$$

$$v(s) = \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s]$$

BELLMAN EQUATIONS

MARKOV DECISION PROCESSES

$$q(s,a) = \mathbb{E}[G_t \mid S_t = s, A_t = a]$$

$$= \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \mid S_t = s, A_t = a]$$

$$= \mathbb{E}[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \dots) \mid S_t = s, A_t = a]$$

$$= \mathbb{E}[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a]$$

$$q(s,a) = \mathbb{E}[R_{t+1} + \gamma q(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a]$$

STATE-VALUE FUNCTION

MARKOV DECISION PROCESSES

$$v(s) = \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s]$$

$$v = \mathcal{R} + \gamma \mathcal{P} v$$

$$(I - \gamma \mathcal{P}) v = \mathcal{R}$$

$$v = (I - \gamma \mathcal{P})^{-1} \mathcal{R}$$

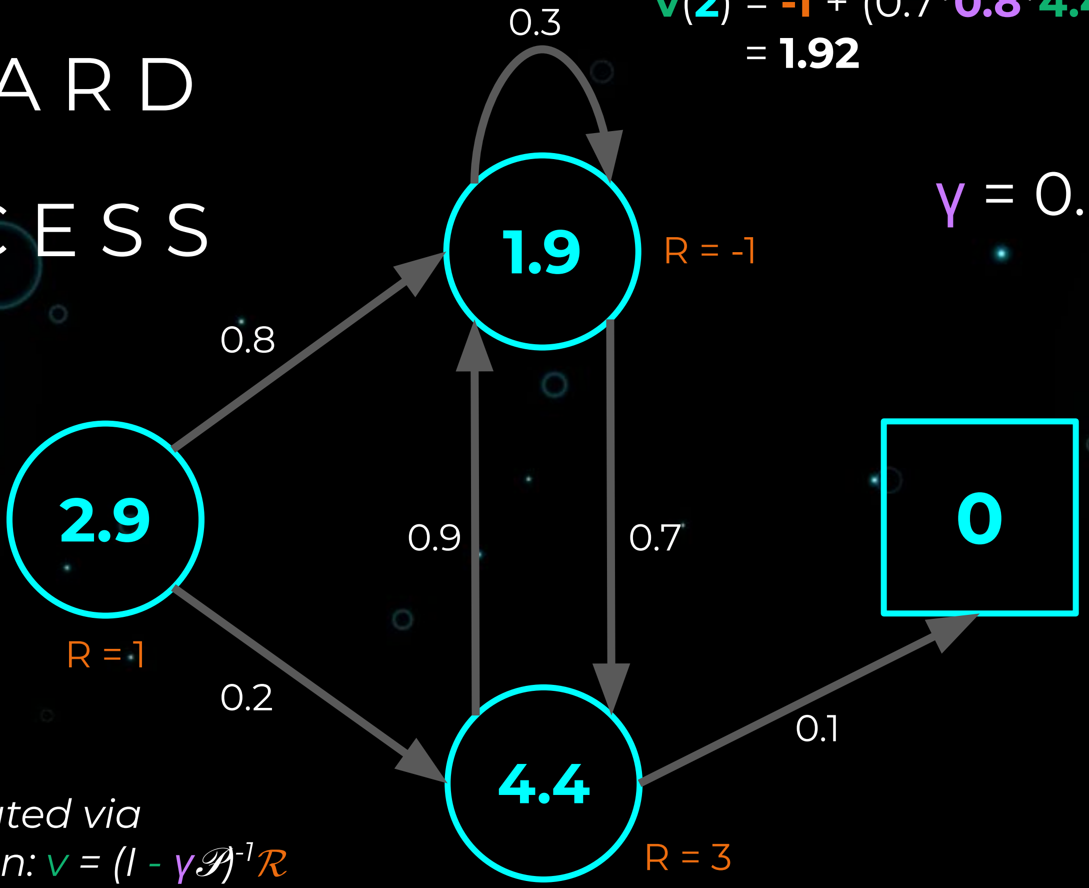
MARKOV REWARD PROCESS

Example manual calculation:

$v(2) = -1 + (0.7*\mathbf{0.8}*\mathbf{4.4}) + (0.3*\mathbf{0.8}*\mathbf{1.9})$

$= \mathbf{1.92}$

$\gamma = 0.8$



***Estimates calculated via closed-form solution:* $v = (I - \gamma\mathcal{P})^{-1}\mathcal{R}$

BELLMAN EQUATIONS

MARKOV DECISION PROCESSES

$$v(s) = \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s]$$

$$q(s,a) = \mathbb{E}[R_{t+1} + \gamma q(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a]$$

MARKOV DECSION PROCESSES

WE FINALLY ARRIVE

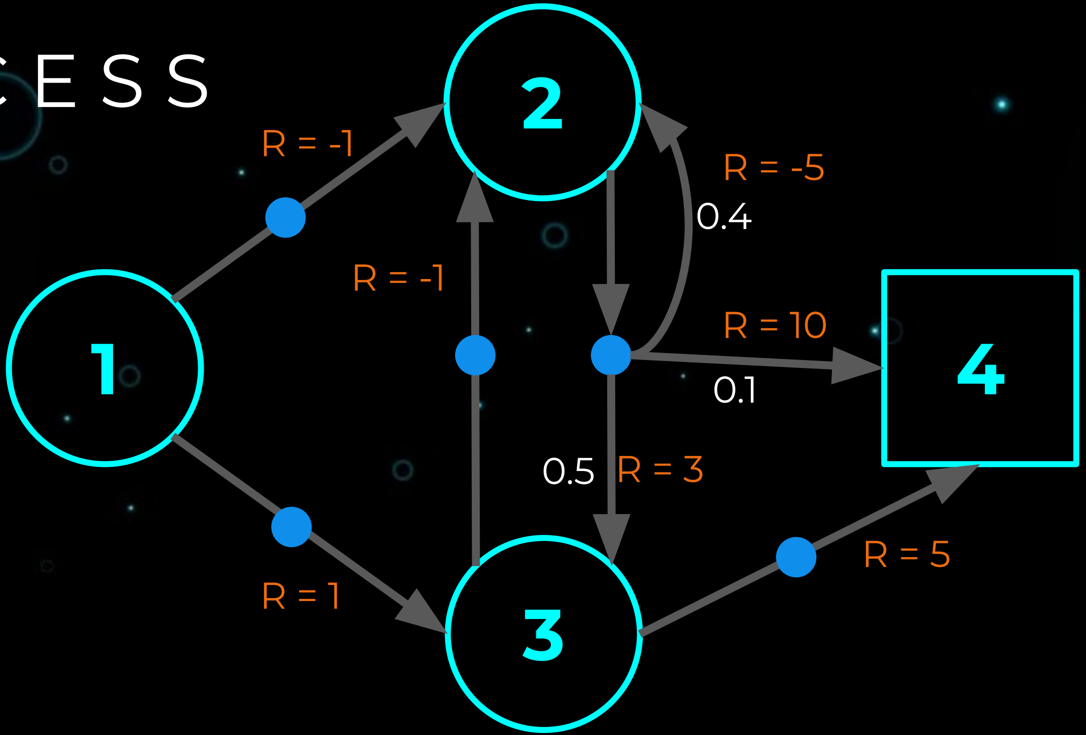
A Markov Decision Process is a tuple $(\mathscr{S}, \mathscr{P}, \mathcal{R}, \gamma, \mathscr{A})$ where:

- $\mathscr{S}$ is a finite set of states.
- $\mathscr{A}$ is a finite set of actions.
- $\mathscr{P}$ is our state-transition probability matrix.

$$\mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$$

- $\mathcal{R}$ is a reward function $\mathcal{R}_s = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$
- γ is a discount factor $\gamma \in [0,1]$

MARKOV DECISION PROCESS



(BEHAVIOURAL) POLICY

A distribution over actions, given states. **Policies *fully define* the behaviour of an agent.** Policies are how an agent chooses actions - decides how to behave - in each state.

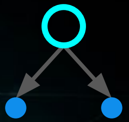
$$\pi(a|s) = \mathbb{P}[A_t = a \mid S_t = s]$$

The probability of taking action a given state s.

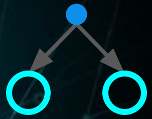
i.e.,

The <u>**probability**</u> of taking action a when in state s while following policy π

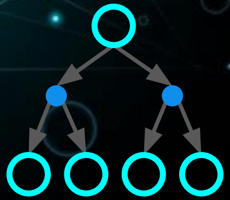
THE BELLMAN EQUATIONS



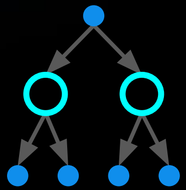
$$v_\pi(s) = \sum \pi(a|s) \cdot q_\pi(s,a)$$



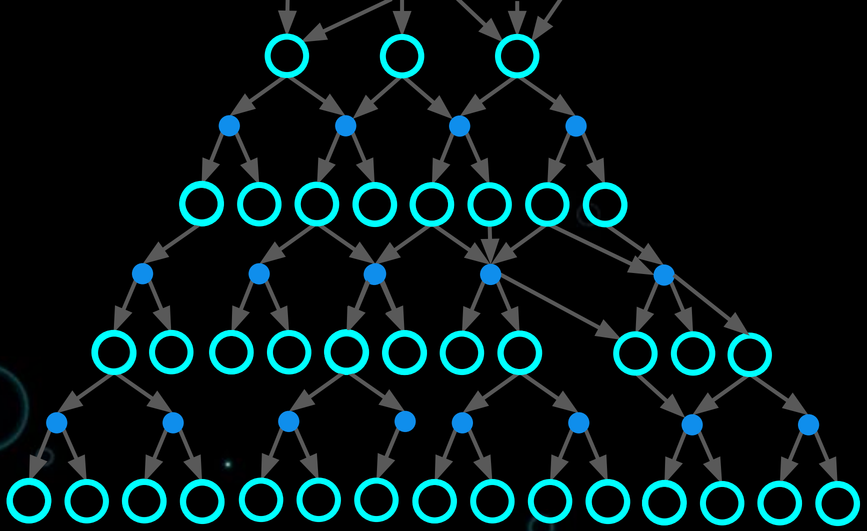
$$q_\pi(s,a) = \sum p(s',r|s,a) \cdot [r + \gamma v_\pi(s')]$$



$$v_\pi(s) = \sum \pi(a|s) \cdot \sum p(s',r|s,a) \cdot [r + \gamma v_\pi(s')]$$



$$q_\pi(s,a) = \sum p(s'|s,a) \cdot [r_{s',a} + \gamma \sum \pi(a'|s') \cdot q_\pi(s',a')]$$



STATE-VALUE FUNCTION

Bellman Equation

The transition-probability-averaged immediate reward plus the discounted future value of the successor states, weighted by the policy-determined action selection probabilities given the current state.

$$v_\pi(s) = \sum_a \pi(a|s) \cdot \sum_{s'} p(s',r|s,a) \cdot [r + \gamma v_\pi(s')]$$

STATE-VALUE FUNCTION

Bellman Equation

Sum over all available actions

The **probability** of taking action a when in state s while following policy π

The reward r you get for taking action a from state s.

Discount factor gamma

$$v_\pi(s) = \sum_a \pi(a|s) \cdot \sum_{s',r} p(s',r|s,a) \cdot [r + \gamma v_\pi(s')]$$

The expected discounted reward earned from the environment, if the agent starts in state s and makes decisions according to policy π thereafter.

Sum over all successor states s' stemming from each state-action pair.

The **probability** of landing in state s' (and getting reward r as a result) if you take action a while in state s.

The expected discounted reward earned from the environment if the agent starts in state s' and follows policy π thereafter.

ACTION-VALUE FUNCTION

Bellman Equation

The transition-probability-averaged sum of the immediate rewards for taking action a from state s and the discounted policy-averaged future value of next selecting action a' from successor state s'.

$$q_\pi(s,a) = \sum_{s'} p(s'|s,a) \cdot [r_{s,a} + \gamma \sum_{a'} \pi(a'|s') \cdot q_\pi(s',a')]$$

ACTION-VALUE FUNCTION

Bellman Equation

The **probability** of landing in state s’ if you take action a from state s.

The reward r you get for taking action a from state s, IF you end up heading to successor state s’.

Sum over all available actions from successor state s’

The **probability** of taking action a’ when in state s’ while following policy π

$$q_{\pi}(s,a) = \sum_{s'} p(s'|s,a) \cdot [r_{s,a} + \gamma \sum \pi(a'|s') \cdot q_{\pi}(s',a')]$$

The expected discounted reward earned from the environment, if the agent takes action a from state s and makes decisions according to policy π thereafter.

Sum over all successor states s’ stemming from each state-action pair.

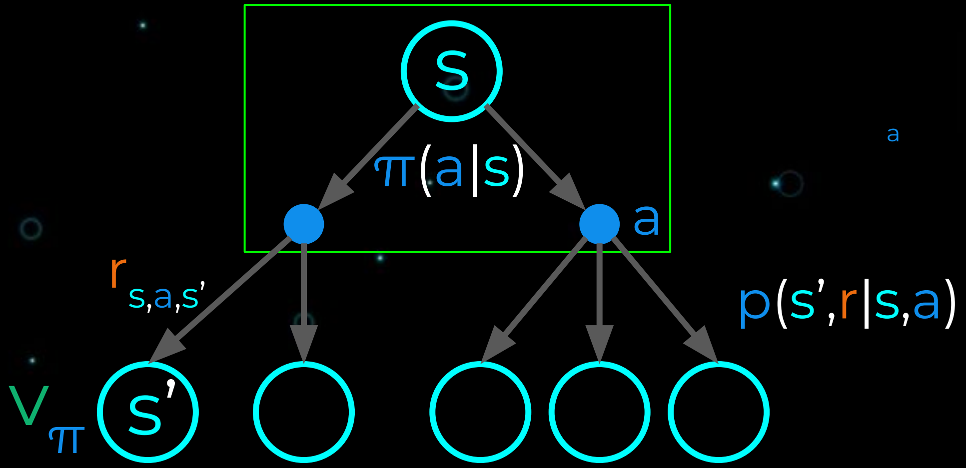
Discount factor gamma

The expected discounted reward earned from the environment, if the agent takes action a’ from state s’ and makes decisions according to policy π thereafter.

STATE-VALUE FUNCTION

Bellman Equation and Backup Diagram

$$v_\pi(s) = \sum_{a} \pi(a|s) \cdot \sum_{s,} p(s',r|s,a) \cdot [r + \gamma v_\pi(s')]$$



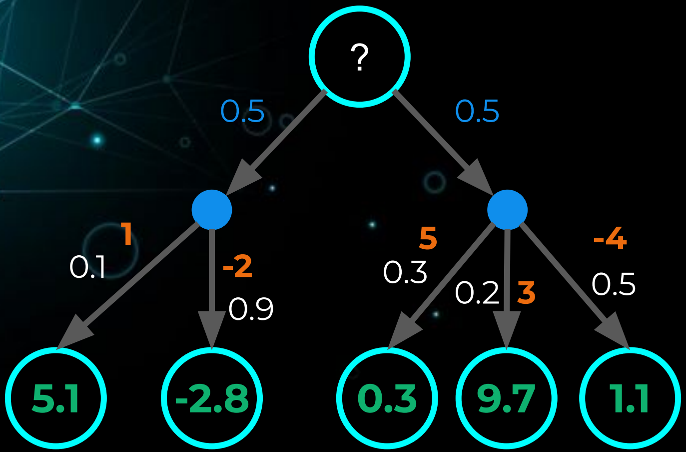
a

$$\boxed{v_\pi(s) = \sum \pi(a|s) \cdot q_\pi(s,a)}$$

STATE-VALUE FUNCTION

Backup Computation Example (Uniform Random Policy)

$$v_\pi(s) = \sum_a \pi(a|s) \cdot \sum_{s',r} p(s',r|s,a) \cdot [r + \gamma v_\pi(s')]$$

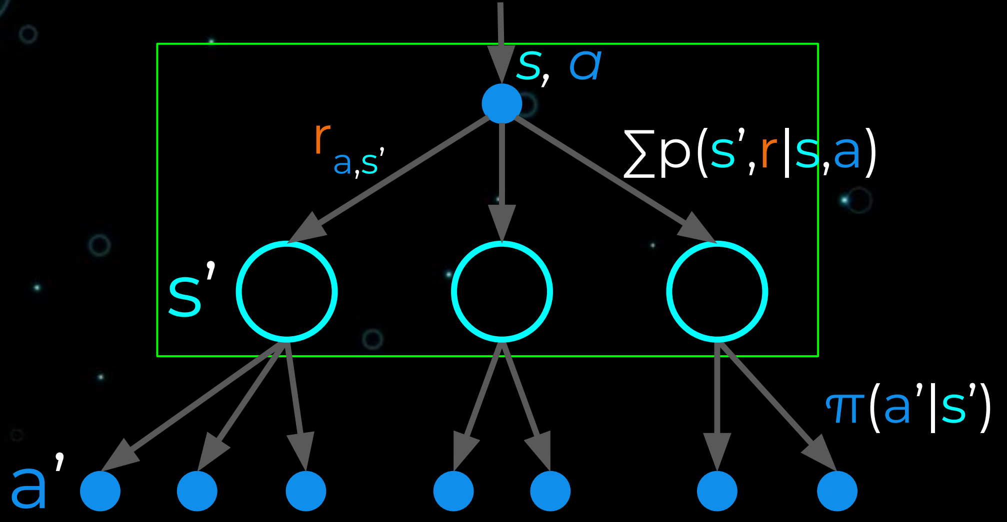


$$(0.5 \cdot 0.1 \cdot [1 + 0.7 \cdot 5.1])$$
$$+$$
$$(0.5 \cdot 0.9 \cdot [-2 + 0.7 \cdot -2.8])$$
$$+$$
$$(0.5 \cdot 0.3 \cdot [5 + 0.7 \cdot 0.3])$$
$$+$$
$$(0.5 \cdot 0.2 \cdot [3 + 0.7 \cdot 9.7])$$
$$+$$
$$(0.5 \cdot 0.6 \cdot [-4 + 0.7 \cdot 1.1])$$

ACTION-VALUE FUNCTION

Bellman Equation and Backup Diagram

$$q_\pi(s,a) = \sum p(s'|s,a) \cdot [r_{a,s'} + \gamma \sum \pi(a'|s') \cdot q_\pi(s',a')]$$

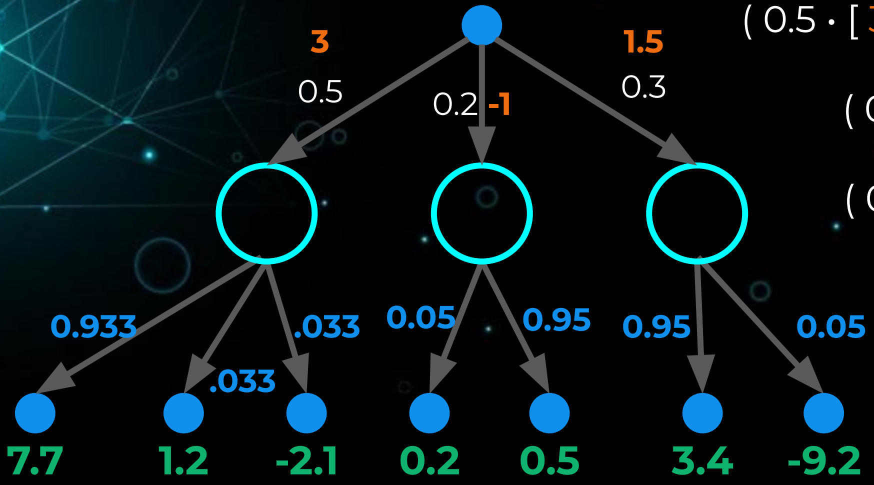


$$\boxed{q_\pi(s,a) = \sum p(s',r|s,a) \cdot [r + \gamma v_\pi(s')]}$$

ACTION-VALUE FUNCTION

Backup Computation Example (Epsilon-Greedy Policy, ε = 0.1)

$$q_\pi(s,a) = \sum p(s'|s,a) \cdot [r_{a,s'} + \gamma \sum \pi(a'|s') \cdot q_\pi(s',a')]$$



$$(0.5 \cdot [3 + 0.7 \cdot ((0.933 \cdot 7.7) + (.033 \cdot 1.2) + (.033 \cdot -2.1))])$$

$$+$$

$$(0.2 \cdot [-1 + 0.7 \cdot ((0.05 \cdot 0.2) + (0.95 \cdot 0.5))])$$

$$+$$

$$(0.3 \cdot [1.5 + 0.7 \cdot ((0.95 \cdot 3.4) + (0.05 \cdot -9.2))])$$

$$=$$

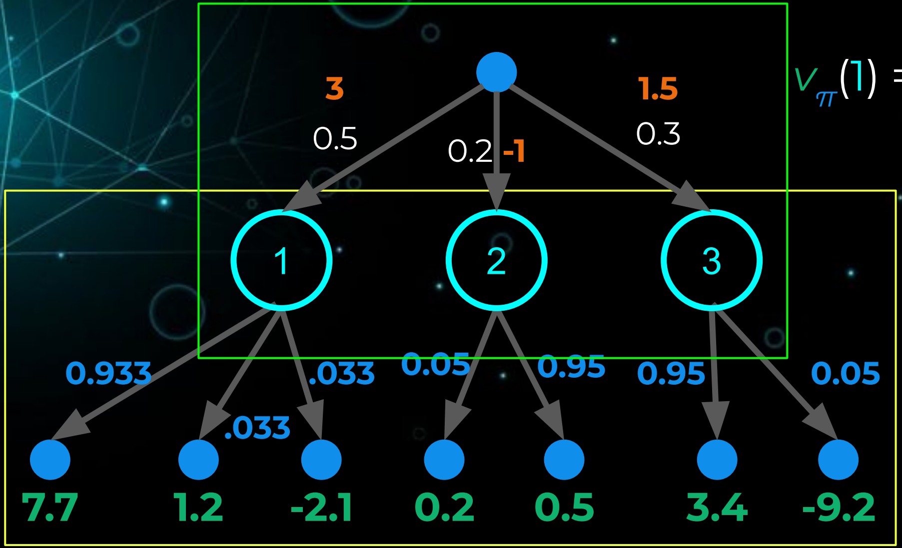
4.81

ACTION-VALUE FUNCTION

GET READY FOR EXTREME ARITHMETIC

$$v_\pi(s) = \sum \pi(a|s) \cdot q_\pi(s,a)$$

$$q_\pi(s,a) = \sum p(s',r|s,a) \cdot [r + \gamma v_\pi(s')]$$



$$v_\pi(1) = (0.933 \cdot 7.7) + (.033 \cdot 1.2) + (.033 \cdot -2.1) = 6.8871$$

$$v_\pi(2) = (0.05 \cdot 0.2) + (0.95 \cdot 0.5) = 0.485$$

$$v_\pi(3) = (0.95 \cdot 3.4) + (0.05 \cdot -9.2) = 2.77$$

$$q_\pi(s,a) =$$
$$(\ 0.5 \cdot [3 + 0.7 \cdot 6.8871]\ ) +$$
$$(\ 0.2 \cdot [-1 + 0.7 \cdot 0.485\ ) +$$
$$(\ 0.3 \cdot [1.5 + 0.7 \cdot 2.77\ ) =$$

4.81