UCL ARTIFICIAL
INTELLIGENCE SOCIETY

TUTORIAL #11

Session 11

Reinforcement Learning

UCL ARTIFICIAL
INTELLIGENCE SOCIETY

LECTURE OVERVIEW

01 **Intro to RL**

What is it?
Why it is so important?

02 **RL Math**

Understanding of the fundamental math background

03 **Policy gradient algorithms**

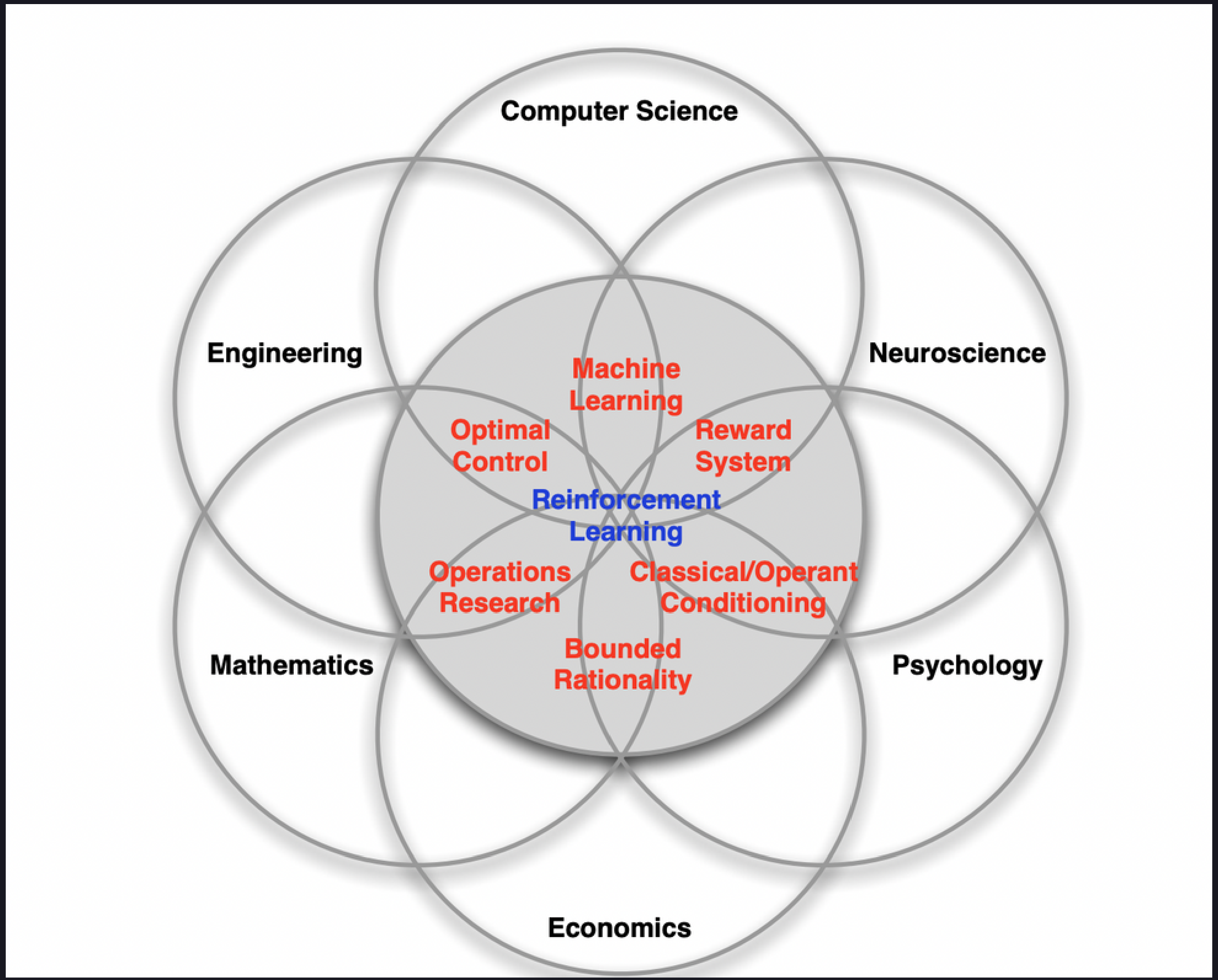
04 **Implementation**

Using RL in practice.

Theory

INTRO TO RL

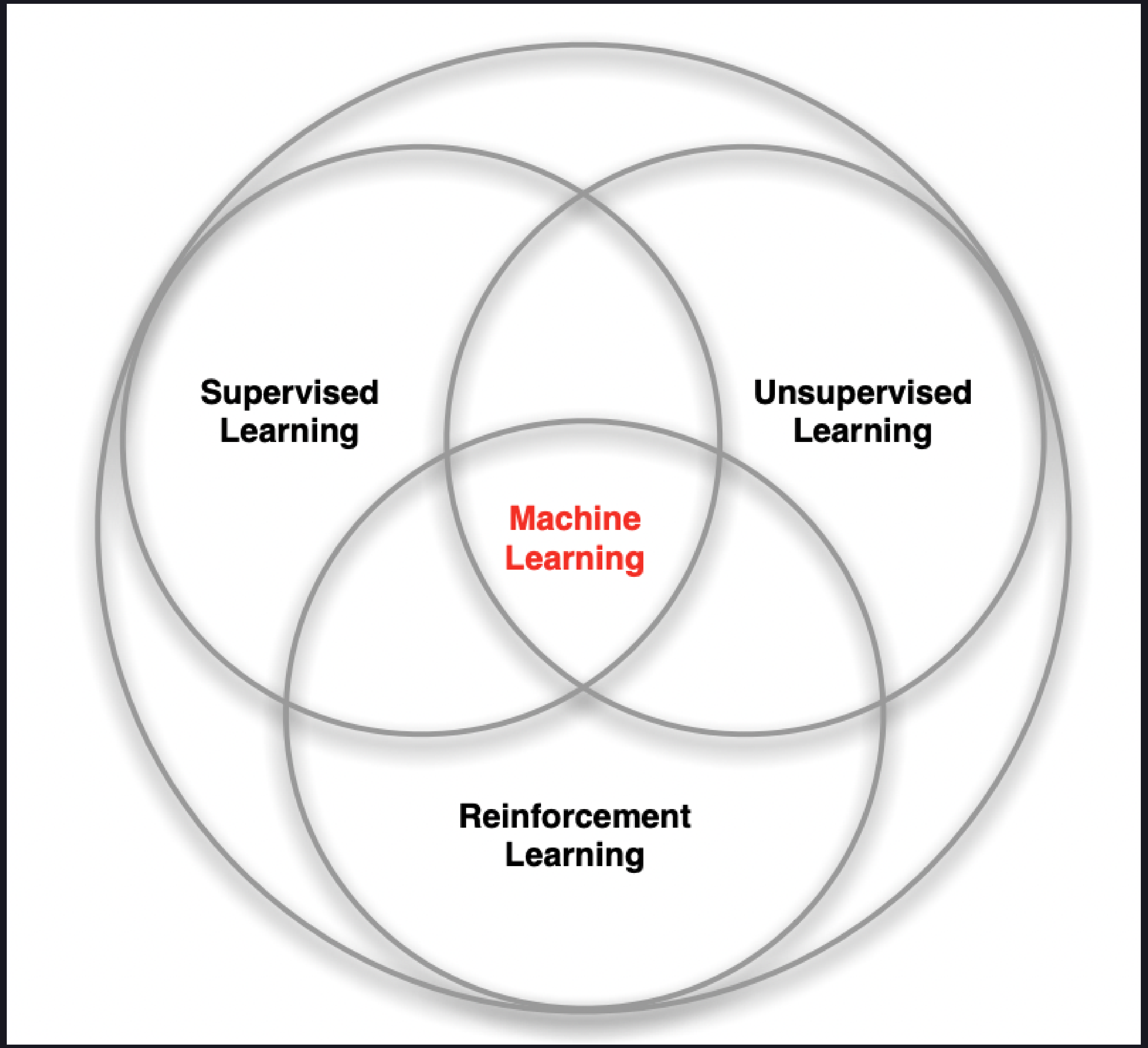
What is RL?



Theory

INTRO TO RL

What differentiates it?



Theory

INTRO TO RL

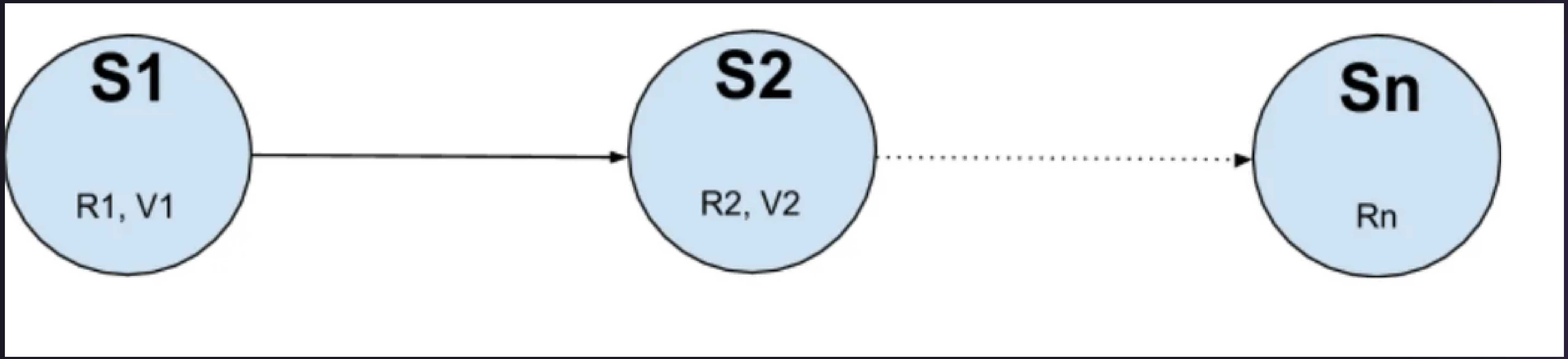
$$V_{\pi}(s) = \sum_{a} \pi(a|s) \sum_{s' \in S} \sum_{r \in R} p(s', r \mid s, a) \left(r + \gamma V_{\pi}(s')\right)$$

Value function of Reinforcement Learning

Theory

INTRO TO RL

States and Rewards



Theory

INTRO TO RL

States and Rewards

$$V(s) = \sum_t \gamma^t R(S_t)$$

Theory

INTRO TO RL

Policy Gradient Algorithm

The reward function is defined as:

$$J(\theta) = \sum_{s \in \mathcal{S}} d^\pi(s) V^\pi(s) = \sum_{s \in \mathcal{S}} d^\pi(s) \sum_{a \in \mathcal{A}} \pi_\theta(a|s) Q^\pi(s, a)$$

Theory

INTRO TO RL

Proof

$$
\begin{aligned}
&\nabla_\theta V^\pi(s) \\
=&\nabla_\theta \Big(\sum_{a\in\mathcal{A}} \pi_\theta(a|s)Q^\pi(s,a)\Big) \\
=&\sum_{a\in\mathcal{A}} \Big(\nabla_\theta \pi_\theta(a|s)Q^\pi(s,a) + \pi_\theta(a|s)\nabla_\theta Q^\pi(s,a)\Big) \\
=&\sum_{a\in\mathcal{A}} \Big(\nabla_\theta \pi_\theta(a|s)Q^\pi(s,a) + \pi_\theta(a|s)\nabla_\theta \sum_{s',r} P(s',r|s,a)(r + V^\pi(s'))\Big) \\
=&\sum_{a\in\mathcal{A}} \Big(\nabla_\theta \pi_\theta(a|s)Q^\pi(s,a) + \pi_\theta(a|s)\sum_{s',r} P(s',r|s,a)\nabla_\theta V^\pi(s')\Big) \\
=&\sum_{a\in\mathcal{A}} \Big(\nabla_\theta \pi_\theta(a|s)Q^\pi(s,a) + \pi_\theta(a|s)\sum_{s'} P(s'|s,a)\nabla_\theta V^\pi(s')\Big)
\end{aligned}
$$

$$
\nabla_\theta V^\pi(s) = \sum_{a\in\mathcal{A}} \Big(\nabla_\theta \pi_\theta(a|s)Q^\pi(s,a) + \pi_\theta(a|s)\sum_{s'} P(s'|s,a)\nabla_\theta V^\pi(s')\Big)
$$

THANK YOU 😎

Theory

RL IMPLEMENTATION

Build and train a simple agent

Theory

RL IMPLEMENTATION

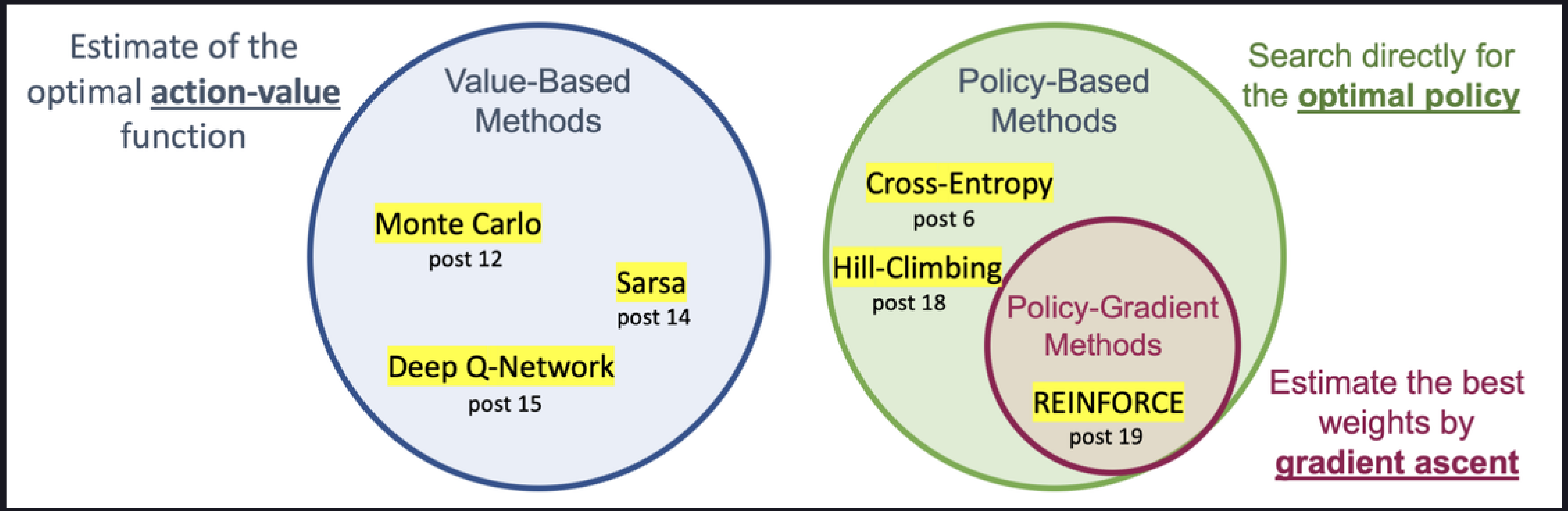
REINFORCE method

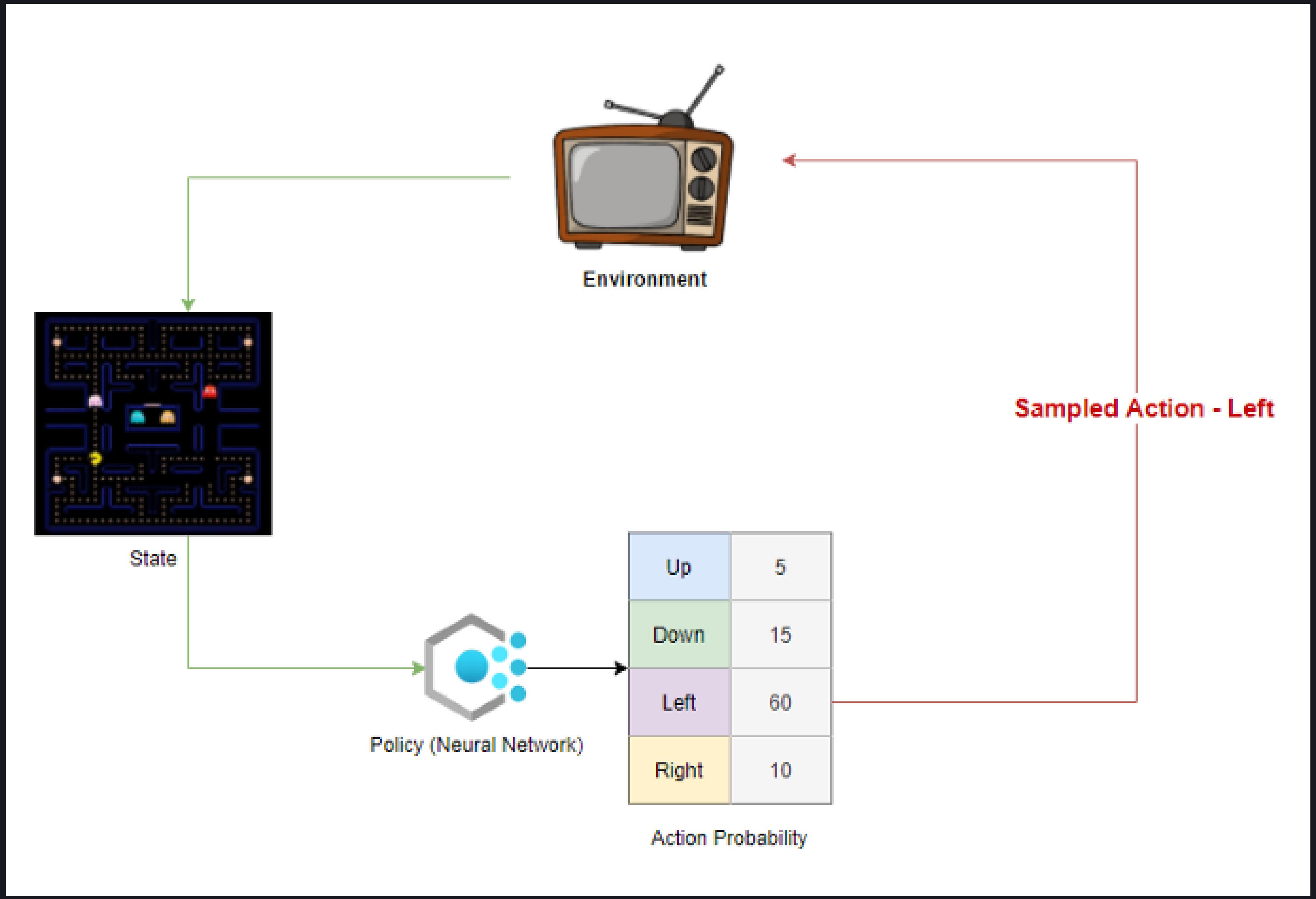
- REINFORCE is a policy gradient method in RL that updates policy based on expected rewards.
- It computes gradient of expected cumulative reward with respect to policy parameters.
- It is computationally efficient and used to learn stochastic/deterministic policies, but suffers from high variance and slow convergence.

Theory

RL IMPLEMENTATION

REINFORCE method





Theory

RL IMPLEMENTATION

Actor–critic reinforcement learning

- Actor-critic RL is a type of algorithm that combines both value-based and policy-based methods in reinforcement learning.
- It uses an actor to learn an optimal policy for decision making and a critic to estimate the value function associated with the policy.
- The actor learns to improve its policy by using the feedback from the critic, which helps to optimize the decision-making process over time.

Theory

RL IMPLEMENTATION

Actor–Critic

Estimates action probability

Estimates value for action

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\, G_t\right] && \text{REINFORCE} \\
&= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\ Q^w(s,a)\right] && \text{Q Actor-Critic} \\
&= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\ A^w(s,a)\right] && \text{Advantage Actor-Critic} \\
&= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\ \delta\right] && \text{TD Actor-Critic}
\end{aligned}
$$

SEE YOU NEXT TIME