Actor Critic Methods: From Paper to Code

Review of Fundamental Concepts

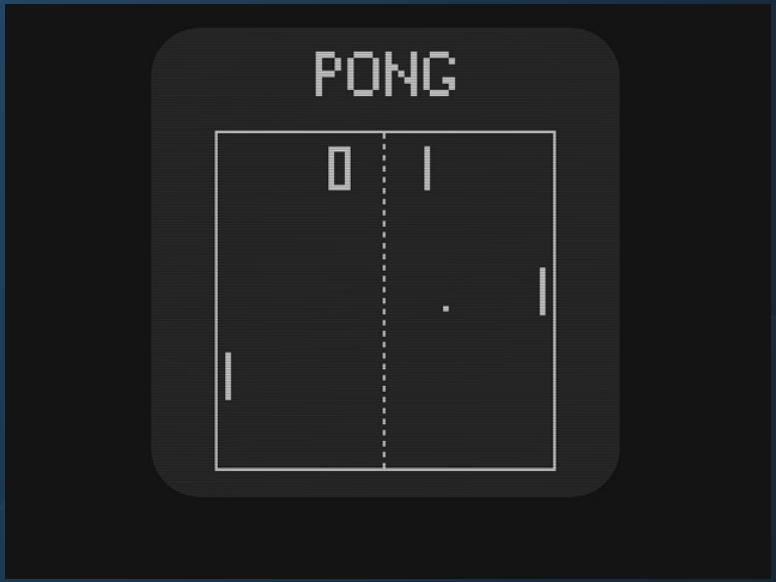
Agent, Environment, Action

Markov Decision Process

State depends only on previous state and action

<u>Markov Decision Process</u>

$$(S_1, A_1, R_1, S_2, A_2, R_2, \ldots)$$



Episodic Returns

These states have value

Present rewards worth more

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Sum of discounted rewards → Episode return

Reward Discounting

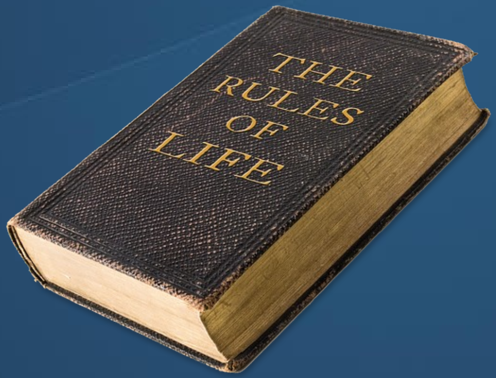
$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \dots$$

$$G_t = R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \dots \right)$$

$$\boxed{G_t = R_{t+1} + \gamma G_{t+1}}$$

But wait… how can we know future rewards?

The Agent's Policy



Mapping of states to actions

Can be probabilistic

Π

Value and Action Value Functions

$$\boxed{v_\pi(s) = E_\pi[G_t | S_t = s] = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s] \, for\ all\ s \in S}$$

$$\boxed{q_\pi(s, a) = E_\pi[G_t | S_t = s, A_t = a] = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a] \, for\ all\ s \in S}$$

Learning from Experience

Interact with environment

Keep track of rewards

Monte Carlo Methods

The Bellman Equation

$$v_\pi(s)=\boxed{E_\pi[G_t|S_t=s]}=E_\pi[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|S_t=s]\, for\, all\, s\in S$$

$$G_t=R_{t+1}+\gamma G_{t+1}$$

$$v_\pi(s)=E_\pi[R_{t+1}+\gamma G_{t+1}|S_t=s]$$

$$v_\pi(s)=\sum_a \pi(a,s)\sum_{s'}\sum_r p(s',r|s,a)[r+\gamma \boxed{E_\pi[G_{t+1}|S_{t+1}=s']]}$$

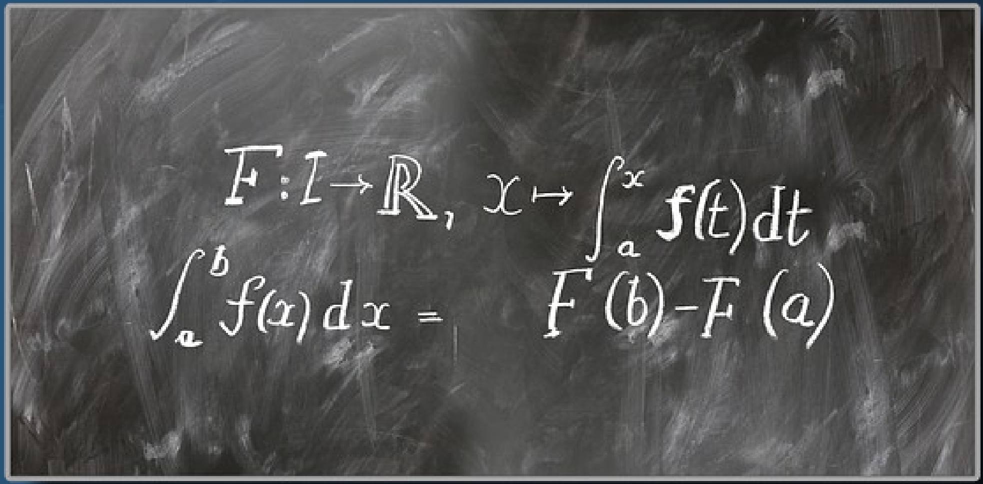
$$\boxed{v_\pi(s)=\sum_a \pi(a,s)\sum_{s',r} p(s',r|s,a)[r+\gamma \boxed{v_\pi(s')]}}$$

<u>Bellman Equation</u>

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

Optimal Policies

Compare policies



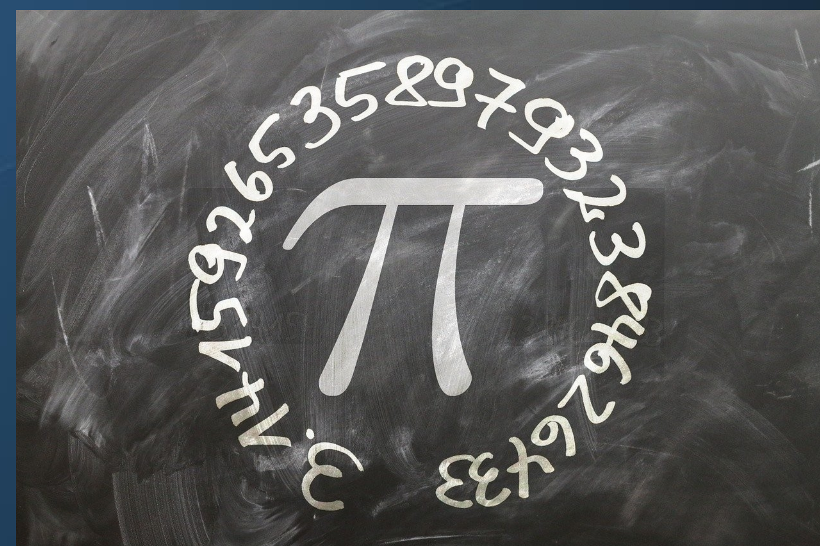
Known dynamics → Model based

Unknown dynamics → Model free

Exploration

Explore Exploit Dilemma

Epsilon greedy (Q Learning)



Approximate Policy Directly

On Policy vs. Off Policy

- One policy generates actions and updates value function → On policy
- One policy generates actions and another policy updates value function → Off policy
- Epsilon greedy → off policy learning
- Policy gradients → on policy learning

Conclusions

- Keep track of rewards to estimate value and action value functions
- Recursive relationship between functions
- Have to interact w/ environment to learn dynamics
- Policy gradient & Actor critic → on policy model free


BALLYS
EIFFEL TOWER RESTAURANT
JERSEY BOYS
Paris
Bellagio