MSiA 490 Lab 4

Fall 2023
Shuyang Wang

Agenda

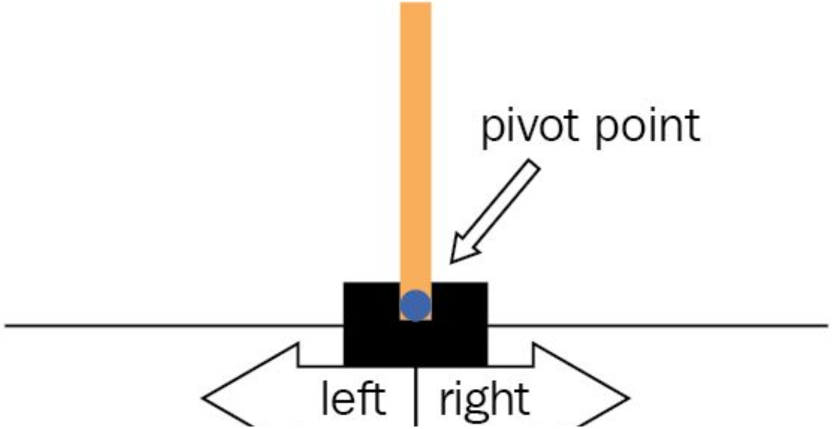
- Announcement:
  - Part 1 of Assignment 1 is posted. Due on Oct. 19th at 10 pm (Thursday).
- Policy Gradient
- CartPole
- Pong
- Implementation

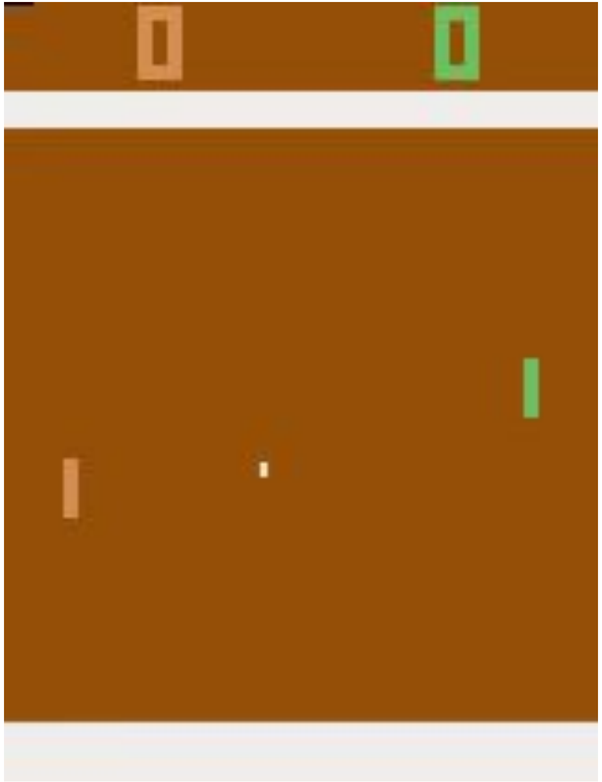
Policy Gradient

$$
\begin{aligned}
\nabla_\theta J(\pi_\theta) &= \nabla_\theta \underset{\tau\sim\pi_\theta}{\mathrm{E}}[R(\tau)] & \\
&= \nabla_\theta \int_\tau P(\tau|\theta)R(\tau) & \text{Expand expectation} \\
&= \int_\tau \nabla_\theta P(\tau|\theta)R(\tau) & \text{Bring gradient under integral} \\
&= \int_\tau P(\tau|\theta)\nabla_\theta \log P(\tau|\theta)R(\tau) & \text{Log-derivative trick} \\
&= \underset{\tau\sim\pi_\theta}{\mathrm{E}}[\nabla_\theta \log P(\tau|\theta)R(\tau)] & \text{Return to expectation form} \\
\therefore \nabla_\theta J(\pi_\theta) &= \underset{\tau\sim\pi_\theta}{\mathrm{E}}\left[\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t|s_t)R(\tau)\right] & \text{Expression for grad-log-prob}
\end{aligned}
$$

CartPole & Pong

- See demo in the notebook.





Training a policy network

Policy(S_t) is a distribution over actions. You can build the distribution using softmax function with logits.

A Neural Network as the Policy Network: Input: State; Output: Logits.

Pseudocode:

1. Sample a batch of B episodes, record the states, actions, and rewards
   Compute reward to go
   For each episode, compute:

      Loss = -1 * sum_t [(Log probability of choosing a_t at state S_t) * (Reward_to_go at t)]

2. Average the loss over B episodes.
3. Take a step in the direction of the Gradient of the loss.