StarAi: Deep Reinforcement Learning

6b: Advantage Actor Critic

William Xu

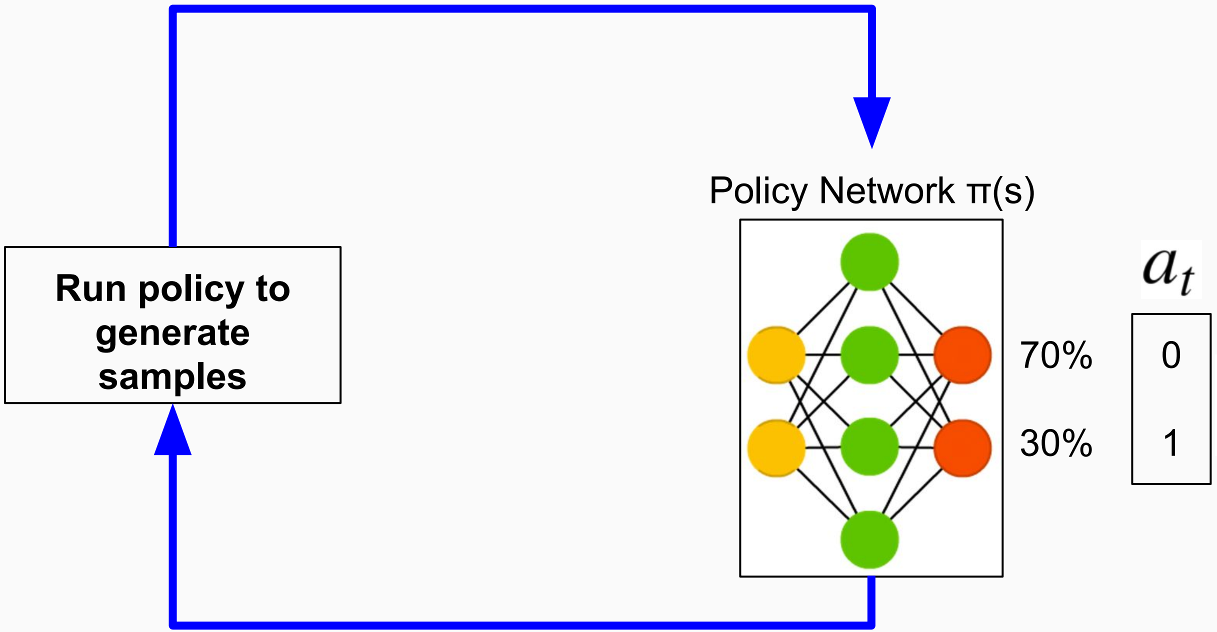
Outline

- Policy gradient update (programming perspective)
- Basic variations of policy gradient update
- Trajectories
- Baselines
- State Value
- Advantage Actor Critic
- A2C

Forms of Policy Gradient

$$\nabla_\theta \log \pi_\theta(a_t \mid s_t)\, \underline{R_t}$$

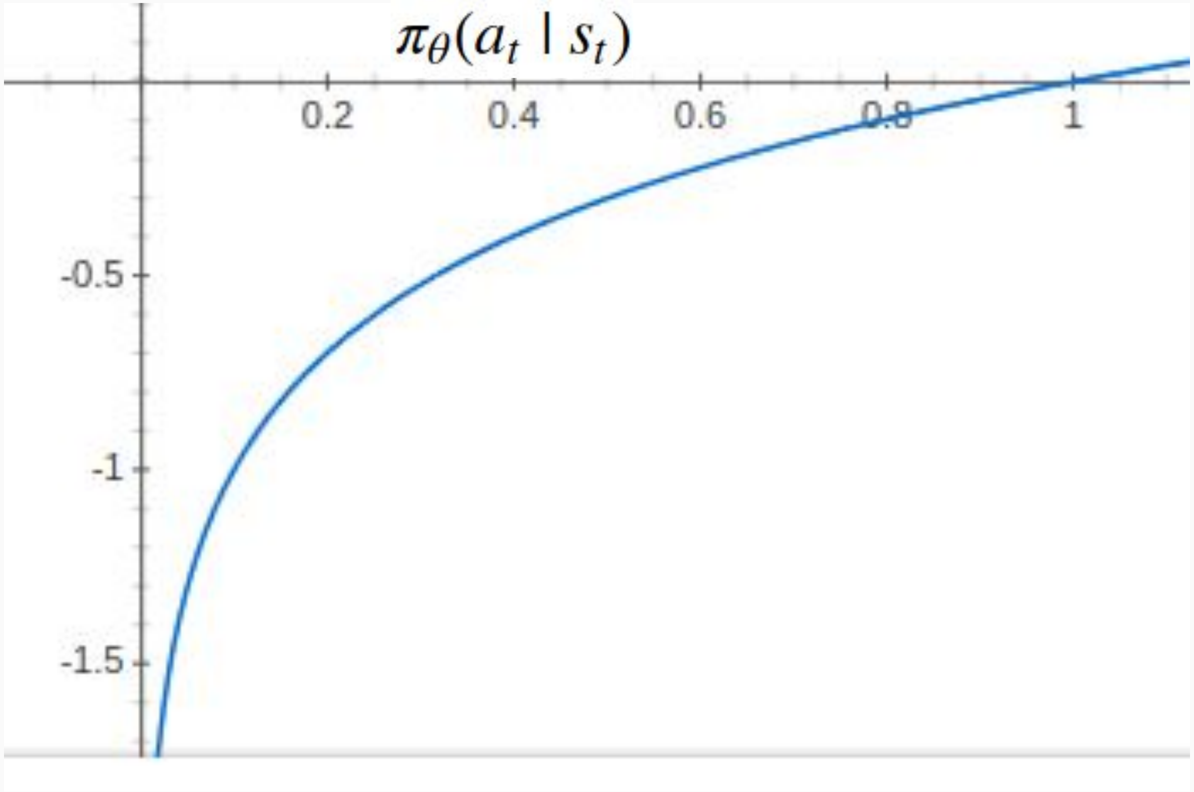
Policy Gradient Update - Coding View



Improve Policy Network according to

$$\nabla_\theta \log \pi_\theta(a_t \mid s_t)\, R_t$$

Log(x)



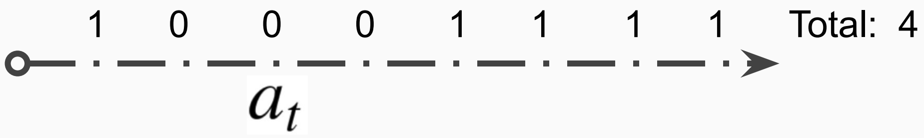
Total Reward of the trajectory

$$\nabla_\theta \log \pi_\theta(a_t \mid s_t) \sum_{t=0}^{\infty} r_t$$

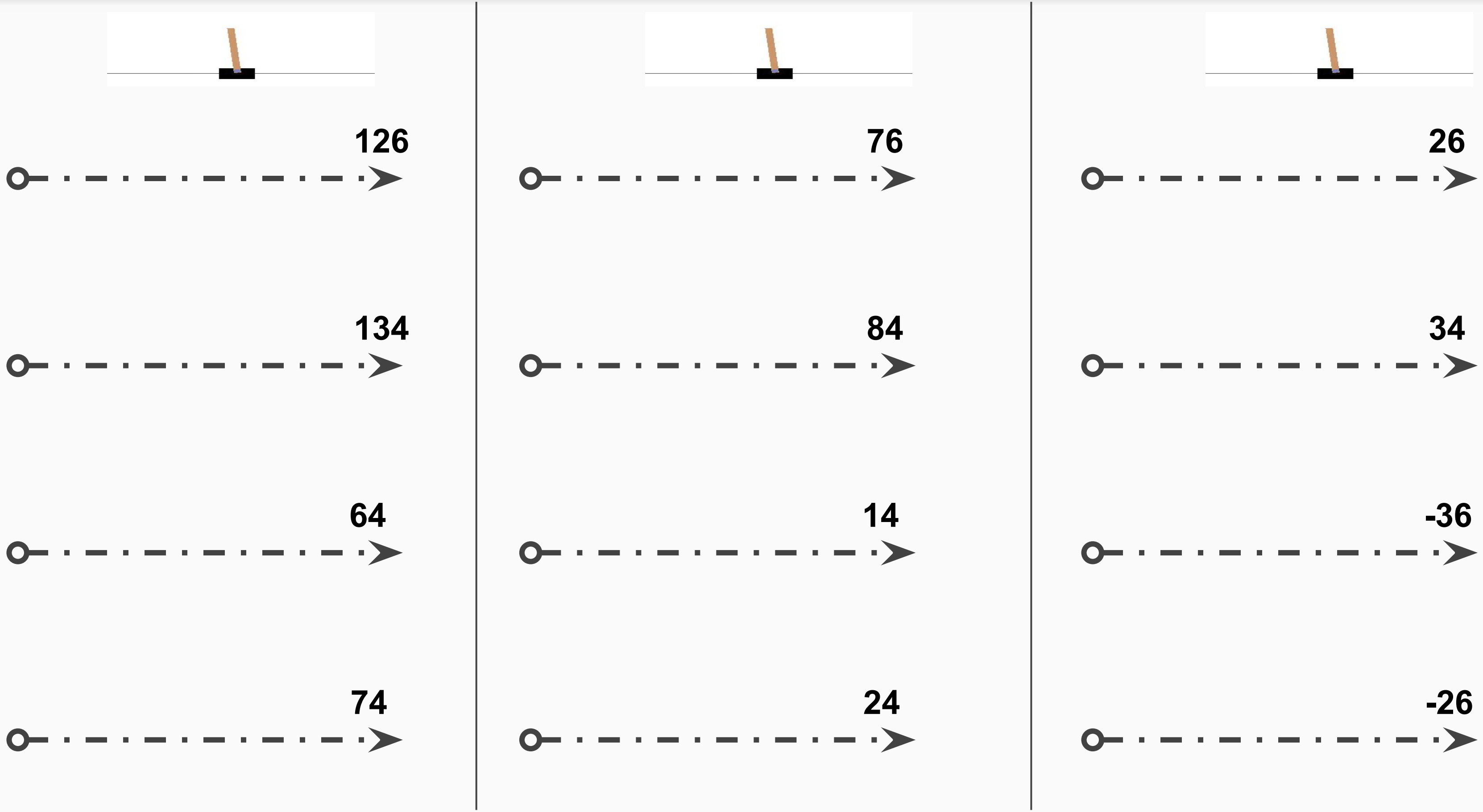
1 0 0 0 1 1 1 1 Total: 5

Reward following action a at time t

$$\nabla_\theta \log \pi_\theta(a_t \mid s_t) \ \sum_{t'=t}^{\infty} r_{t'}$$



Trajectories



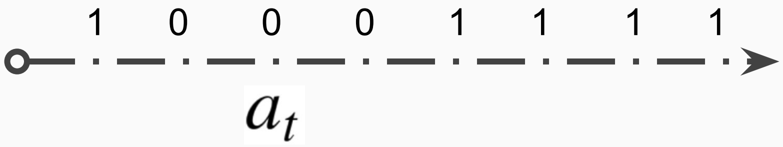
Baseline Options

$$\nabla_\theta \log \pi_\theta(a_t \mid s_t) \left(\sum_{t'=t}^{\infty} r_{t'} - b(s_t)\right)$$

- Some arbitrary baseline appropriate for the env
- Average total reward over all past experiences (if using total rewards)
- Average total reward from time step t (if using total reward from time t)

V(s)

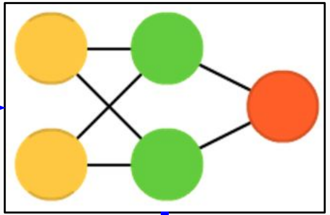
$$\nabla_\theta \log \pi_\theta(a_t \mid s_t) \left(\sum_{t'=t}^{\infty} r_{t'} - V(s_t)\right)$$



Estimate V(s) at time t

V(s) - 1 sample view

Value Network V(s)

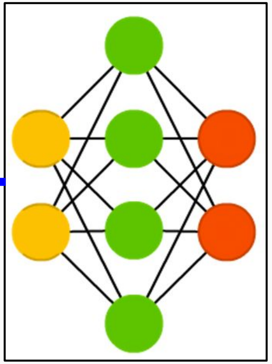


Improve Value Network with loss

$$\frac{1}{2}\left\|(V(s_t) - \textstyle\sum_{t'=t}^{\infty} r_{t'})\right\|^2$$

Run policy to generate samples

Policy Network π(s)



Improve Policy Network according to

$$\nabla_\theta \log \pi_\theta(a_t \mid s_t)\,(\textstyle\sum_{t'=t}^{\infty} r_{t'} - V(s_t))$$

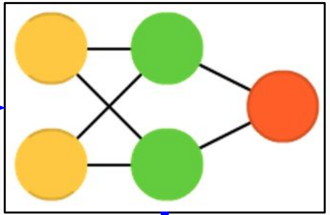
Actor Critic

$$V(s_t) = r_t + \gamma V(s_{t+1})$$

Advantage Actor Critic

Run policy to generate samples

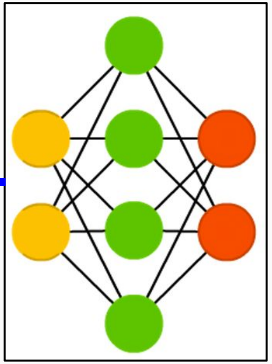
Value Network V(s)



Improve Value Network according to

$$\frac{1}{2}\left\|\left(V(s_t) - \sum_{t'=t}^{\infty} r_{t'}\right)\right\|^2$$

Policy Network π(s)



Improve Policy Network according to

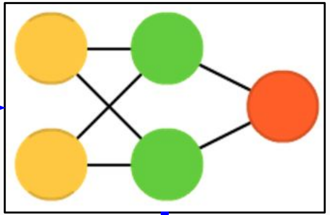
$$A(s_t, a_t) = r_t + \gamma V(s_{t+1}) - V(s_t)$$

$$\nabla_\theta \log \pi_\theta(a_t \mid s_t)\, A(s_t, a_t)$$

Advantage Actor Critic

Run policy to generate samples

Value Network V(s)

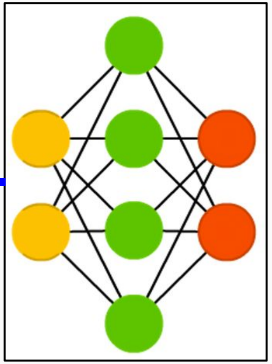


Improve Value Network according to

$$\frac{1}{2}\left\|\left(V(s_t) - \sum_{t'=t}^{\infty} r_{t'}\right)\right\|^2$$

Or use $r_t + \gamma V(s_{t+1})$

Policy Network π(s)



Improve Policy Network according to

$$A(s_t, a_t) = r_t + \gamma V(s_{t+1}) - V(s_t)$$

$$\nabla_\theta \log \pi_\theta(a_t \mid s_t)\, A(s_t, a_t)$$

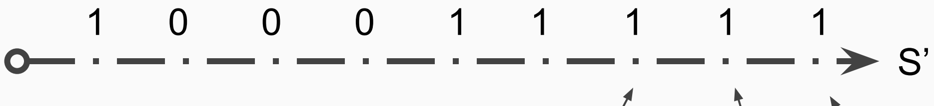
A2C

- A3C: Asynchronous Advantage Actor Critic
  - Part of the paper: Asynchronous Methods for Deep Reinforcement Learning (2016)
  - Multiple agents collecting samples using the same weights
  - A centralised module that learns from the samples and regularly pushes weight updates to the agents
- A2C is without the Async but still multiple agents collecting samples

A2C

- k step bootstrapped advantage A(s, a) estimate where k is upper bound by 5

$$\sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(s_{t+k}) - V(s_t)$$



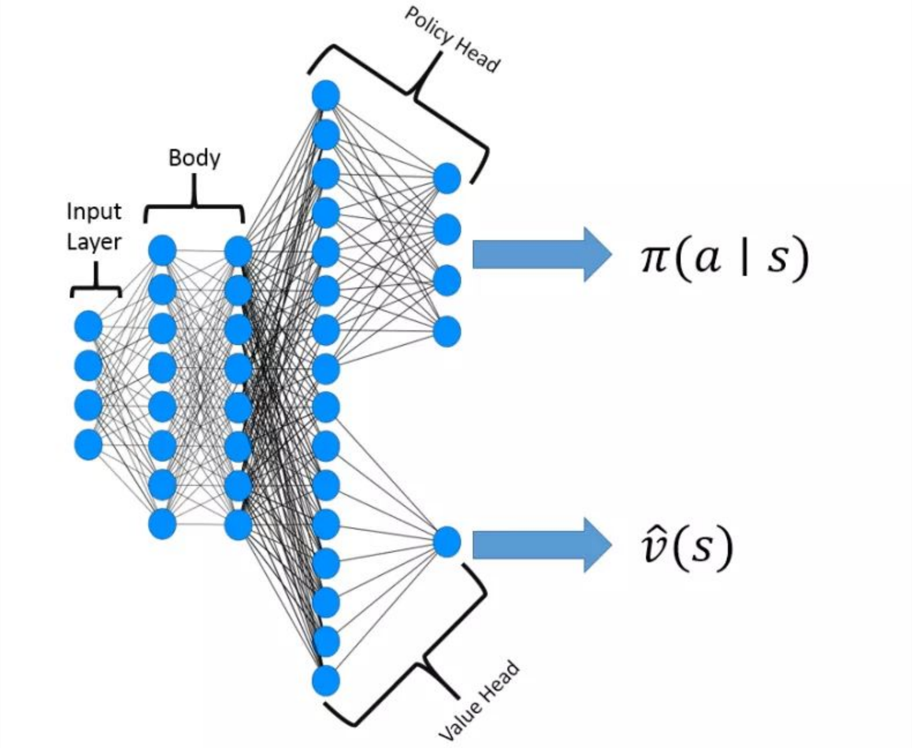
$$1 + \gamma V(s')$$

$$1 + \gamma 1 + \gamma^2 V(s')$$

$$1 + \gamma 1 + \gamma^2 1 + \gamma^3 V(s')$$

A2C

- Shared neural network body



A2C

- Loss function where H is the entropy and beta is 0.01

$$\underbrace{\|R(s_t, a_t) - V(s_t)\|^2}_{\text{Value Loss}} - \underbrace{\log \pi_\theta(a_t \mid s_t)\,(R(s_t, a_t) - V(s_t))}_{\text{Actor Loss}} - \underbrace{\beta H(\pi(s_t))}_{\text{Entropy}}$$

Next

- Checkout Open AI Baselines
- Checkout the PPO paper and the OpenAI implementation