# Implementing Actor-Critic Architecture on Grid World

CAP6629 Reinforcement Learning Project 3

Fanchen Bao

### Introduction

- Tabular method
  - Small state and action space
  - Curse of dimensionality
- Actor-Critic
  - Neural network to simulate state value function and policy function
  - Good for large or continuous state or action space

Parameterize total reward function (and policy function)

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

Policy gradient theorem [1]

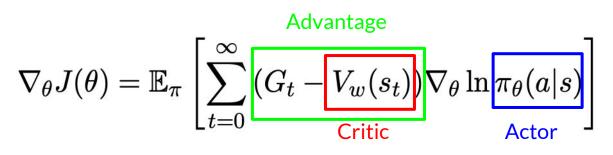
$$egin{aligned} 
abla_{ heta} J( heta) &= 
abla_{ heta} \sum_{s \in S} d^{\pi}(s) \sum_{a \in A} \pi_{ heta}(a|s) Q^{\pi}(s,a) \\ 
abla &= \sum_{s \in S} d^{\pi}(s) \sum_{a \in A} 
abla_{ heta} \pi_{ heta}(a|s) Q^{\pi}(s,a) \\ 
abla &= \mathbb{E}_{\pi} \left[ Q^{\pi}(s,a) \nabla_{ heta} \ln \pi_{ heta}(a|s) \right] \end{aligned}$$

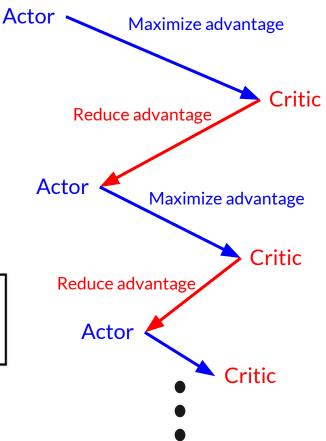
• Policy gradient theorem [2]

Advantage

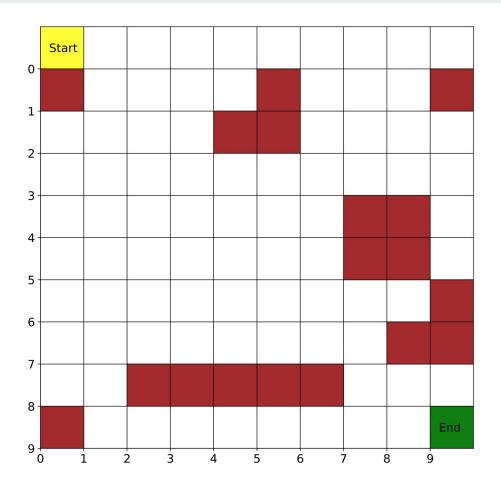
$$abla_{ heta}J( heta) = \mathbb{E}_{\pi} \left[ \sum_{t=0}^{\infty} \left[ G_t - V_w(s_t) \right] 
abla_{ ext{Critic}} 
abla_{ heta} \ln \pi_{ heta}(a|s) 
ight]_{ ext{Actor}}$$

Antagonistic interaction between actor and critic





- Grid World
  - $\circ$  Move:  $\uparrow\downarrow\leftarrow\rightarrow$
  - o Penalty per move: -1
  - Find optimal path

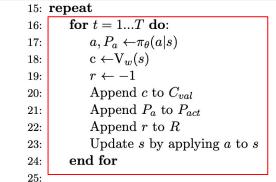


Pseudo-code

$$\circ$$
  $\gamma = 0.9$ 

$$\circ$$
  $\alpha_{\theta} = 0.01$ 

$$\alpha_{\rm w} = 0.01$$



end for

42:  $ep \leftarrow ep + 1$ 43: **until**  $ep = max\_eps$ 

30:

31: 32:

33:

34:

35:

36:

37: 38:

39:

40: 41: ▷ Each step in an episode
 ▷ Sample action and its probability
 ▷ Obtain estimated critic value
 ▷ Obtain the reward, which is always -1

#### Monte Carlo

26:  $g \leftarrow 0$   $\triangleright$  Accumulation of discounted reward 27: **for** r in reversed R **do**:  $\triangleright$  The discounted total reward is computed in reverse 28:  $g \leftarrow r + \gamma g$ 29: Append g to G

Reverse 
$$G$$

$$A \leftarrow G - C_{val} \qquad \qquad \triangleright \text{ Obtain advantage via pair-wise subtraction}$$

$$L_c \leftarrow A^2/2T \qquad \qquad \triangleright \text{ Mean squared error, critic loss}$$

$$L_a \leftarrow -\ln(P_{act}) \cdot A \qquad \qquad \triangleright \text{ Dot product, actor loss}$$

$$Compute \nabla_{\theta}\pi_{\theta} \text{ based on } L_a \qquad \qquad \triangleright \text{ Obtain actor gradient}$$

$$\theta \leftarrow \theta + \alpha_{\theta}\nabla_{\theta}\pi_{\theta} \qquad \qquad \triangleright \text{ Update actor parameters}$$

$$Compute \nabla_w V_w \text{ based on } L_c \qquad \qquad \triangleright \text{ Obtain critic gradient}$$

$$w \leftarrow \mathbf{w} + \alpha_w \nabla_w V_w \qquad \qquad \triangleright \text{ Update critic parameters}$$

Pseudo-code

$$\circ$$
  $y = 0.9$ 

$$\circ$$
  $\alpha_{\theta} = 0.01$ 

$$\alpha_{\rm w} = 0.01$$



20: Append c to  $C_{val}$ 21: Append  $P_a$  to  $P_{act}$ 22: Append r to R

23: Update s by applying a to s

24: **end for** 25:

26:  $g \leftarrow 0$ 27: **for** r in reversed R **do**: 28:  $g \leftarrow r + \gamma g$ 29: Append g to G30: **end for** 31: Reverse G

43: **until**  $ep = max_eps$ 

 $\,\,\triangleright\,$  Accumulation of discounted reward  $\,\triangleright\,$  The discounted total reward is computed in reverse

#### Discounted rewards

32:  $A \leftarrow G - C_{val}$ 33: ▶ Obtain advantage via pair-wise subtraction  $L_c \leftarrow A^2/2T$ 34: ▶ Mean squared error, critic loss  $L_a \leftarrow -\ln(P_{act}) \cdot A$ 35: ▶ Dot product, actor loss 36: Compute  $\nabla_{\theta}\pi_{\theta}$  based on  $L_{a}$ ▷ Obtain actor gradient 37: 38:  $\theta \leftarrow \theta + \alpha_{\theta} \nabla_{\theta} \pi_{\theta}$ ▶ Update actor parameters Compute  $\nabla_w V_w$  based on  $L_c$ ▷ Obtain critic gradient 39:  $w \leftarrow w + \alpha_w \nabla_w V_w$ ▶ Update critic parameters 40: 41:  $ep \leftarrow ep + 1$ 

#### Pseudo-code y = 0.9

$$\alpha_{\theta} = 0.01$$

$$\circ$$
  $\alpha_{\rm w} = 0.01$ 

$$abla_{ extstyle W} extstyle 0.01 \ 
abla_{ extstyle W} J( heta) = \mathbb{E}_{\pi} \left[ \sum_{t=0}^{\infty} \left( G_t - V_w(s_t) 
ight) 
abla_{ heta} \ln \pi_{ heta}(a|s) 
ight]$$

15: repeat 16:

17:

18:

19:

24: 25:

26:

27:

30:

31:

36:

37:

for t = 1...T do:

 $a, P_a \leftarrow \pi_{\theta}(a|s)$  $c \leftarrow V_w(s)$ 

 $r \leftarrow -1$ 

Append  $P_a$  to  $P_{act}$ 

Update s by applying a to s

> Accumulation of discounted reward ▶ The discounted total reward is computed in reverse

▶ Each step in an episode

▶ Sample action and its probability

▷ Obtain the reward, which is always -1

▷ Obtain estimated critic value

Reverse G

Compute  $\nabla_{\theta}\pi_{\theta}$  based on  $L_{a}$ 

Compute  $\nabla_w V_w$  based on  $L_c$ 

 $\theta \leftarrow \theta + \alpha_{\theta} \nabla_{\theta} \pi_{\theta}$ 

 $w \leftarrow w + \alpha_w \nabla_w V_w$ 

end for

Append g to G

 $g \leftarrow \mathbf{r} + \gamma g$ 

for r in reversed R do:

▶ Obtain advantage via pair-wise subtraction ▶ Mean squared error, critic loss

Compute loss ▶ Dot product, actor loss

▷ Obtain actor gradient

▶ Update actor parameters ▷ Obtain critic gradient ▶ Update critic parameters

20: Append 
$$c$$
 to  $C_{val}$   
21: Append  $P_a$  to  $P_{ac}$ 

Append r to R22: 23:

end for

 $g \leftarrow 0$ 

38: 39:

40:

41:  $ep \leftarrow ep + 1$ 

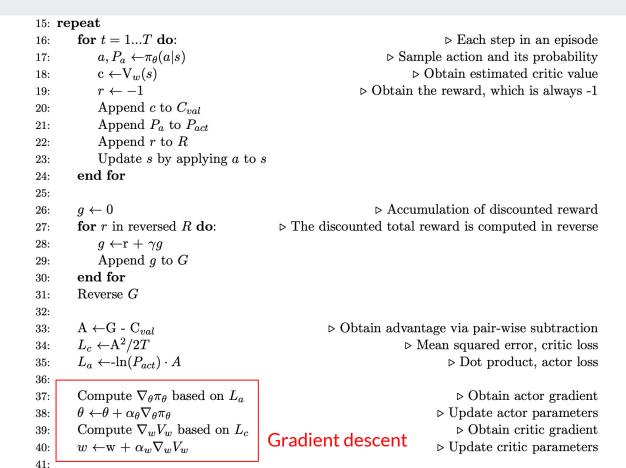
43: **until**  $ep = max_eps$ 

Pseudo-code

$$\circ$$
  $\gamma = 0.9$ 

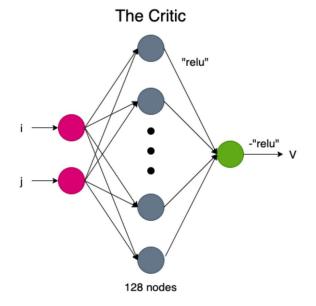
$$\circ$$
  $\alpha_{\theta} = 0.01$ 

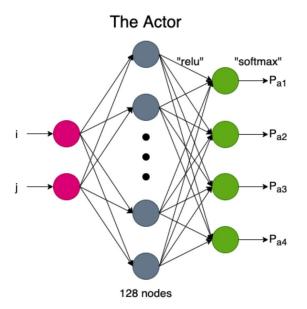
$$\circ$$
  $\alpha_{\rm w} = 0.01$ 



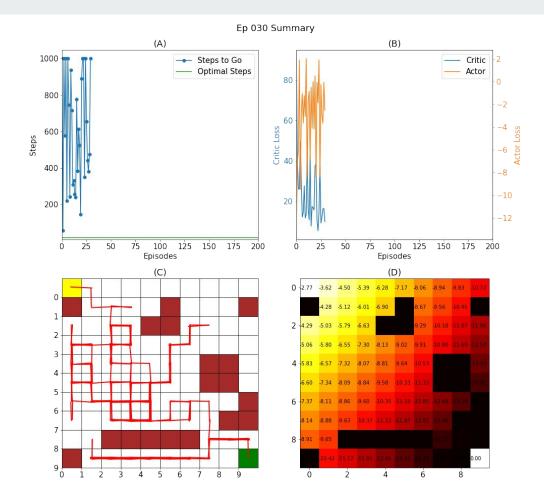
42:  $ep \leftarrow ep + 1$ 43: **until**  $ep = max\_eps$ 

Neural network

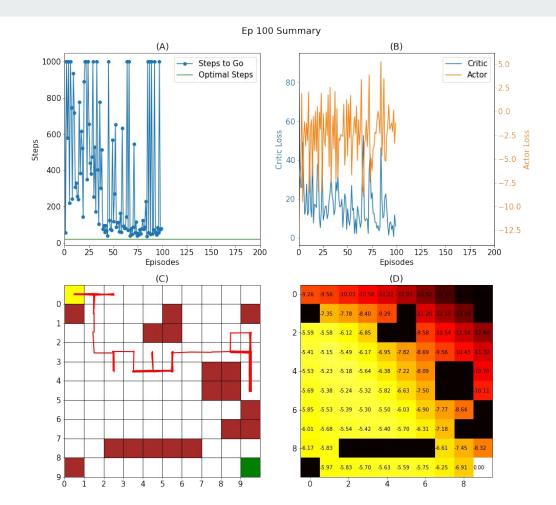




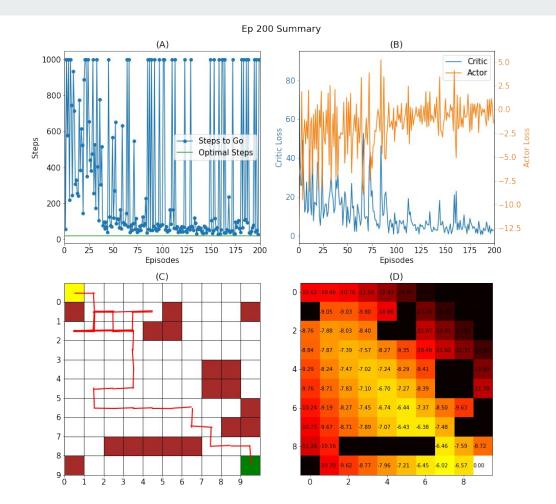
Two openings



Two openings

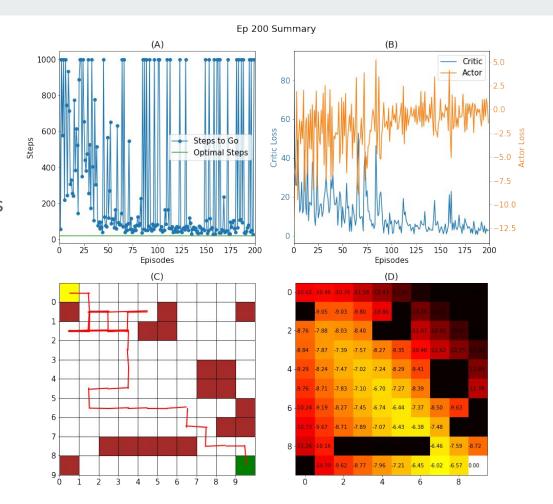


• Two openings

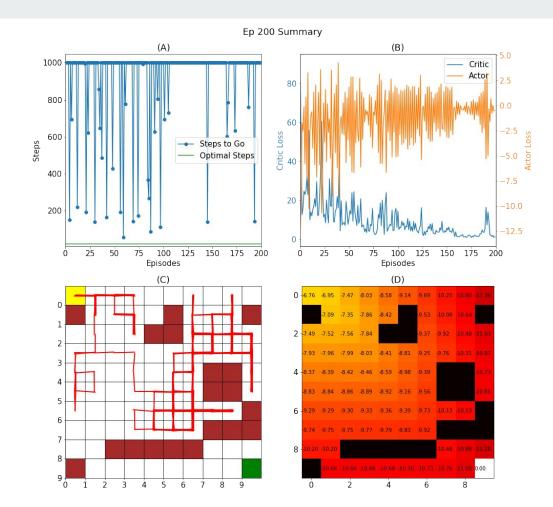


TD-λ **Actor-Critic** (B) **Evaluation** 0.30 - Steps to Go - Critic Optimal Steps 0.25 0.20 Comparison § S 0.15 Optimal Steps with TD-λ 0.10 200-0.05 100 125 150 175 200 50 100 125 150 175 200 40 Episodes Episodes % 9 State Value -10 -12

Only the right opening is used

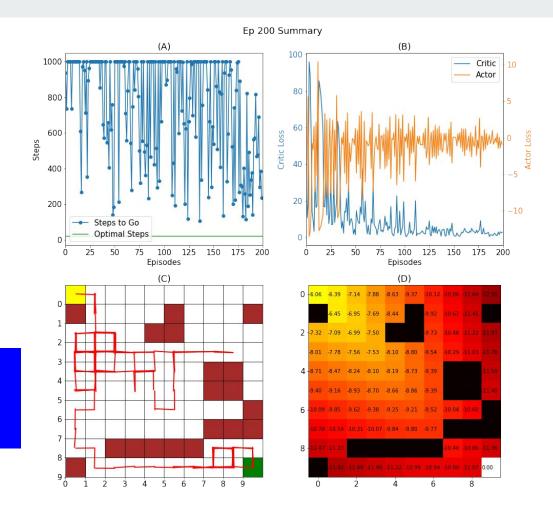


One opening



- One opening
- Bottom right to top left

# Domain knowledge is important!



# **Conclusions**

- Actor-Critic depends on random walk initially.
- Actor-Critic only approximate optimal solution.
- Actor-Critic is slower and less stable in training than TD-λ.
- Domain knowledge crucial in solving reinforcement learning problem.

#### References

- [1] L. Weng, "Policy Gradient Algorithms," Apr. 2018. [Online]. Available: https://lilianweng.github.io/2018/04/08/policy-gradient-algorithms.html
- [2] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," arXiv:1506.02438 [cs], Oct. 2018, arXiv: 1506.02438. [Online]. Available: http://arxiv.org/abs/1506.02438