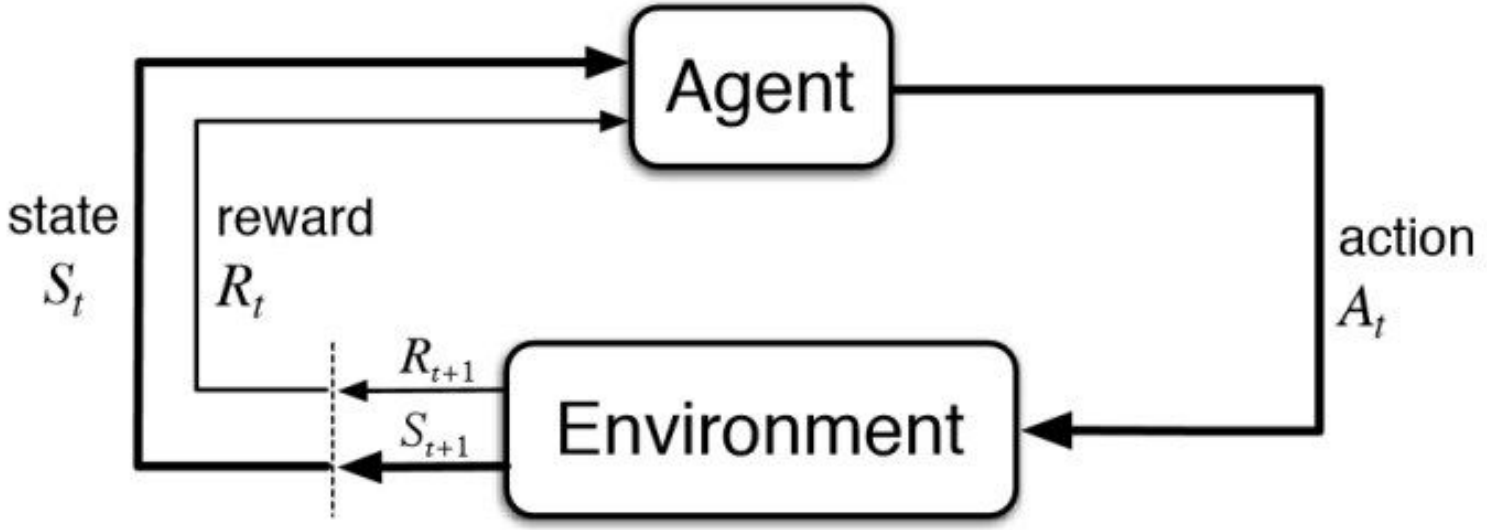
Deep Reinforcement Learning for Sparse Rewards

Fucheng Li and Shelvin Pauly

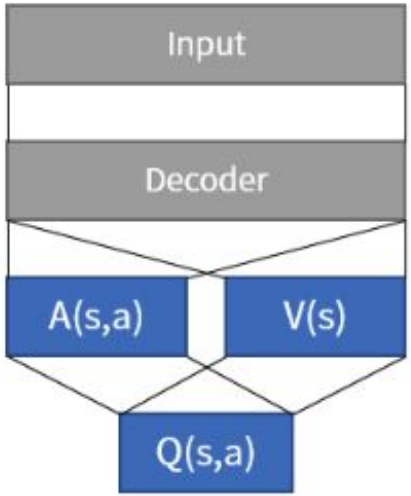
Reinforcement Learning



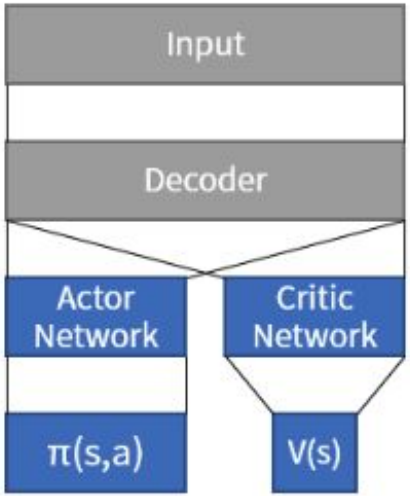
Reinforcement Learning

- Reinforcement as a Markov Decision Process <S, A, R, P, γ>
  - S is the state space
  - A is the action space
  - R is the reward function
  - P is the transition probability function from s to s' when action a is taken
  - γ is the discount factor for rewards
- A policy π is the mapping of S to the probability distribution over A where π(s,a) is the probability of taking action a in state s
- Return of policy π is $J_R(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right]$
- Objective 1: Find the optimal policy to maximize J_R
- Objective 2: Guarantee performance in practical situations (sparse reward settings)

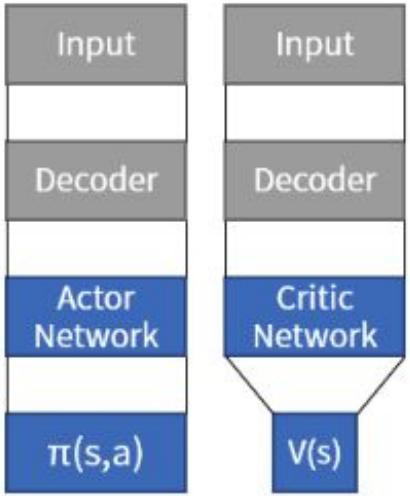
Q-Learning v.s. Actor-Critics



Dueling DQN



Actor-Critic Type #1



Actor-Critic Type #2

Literature Review

- Trust Region Optimization (TRPO) [Schulman et al., 2015]

$$\begin{aligned} &\underset{\theta}{\text{maximize}} \quad \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] \\ &\text{subject to} \quad \hat{\mathbb{E}}_t[\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]] \leq \delta. \end{aligned}$$

- Proximal Policy Optimization [Schulman et al. 2017]

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

- Double Deep Q-Learning (DDQN) [Hasselt et al. 2015]

$$Q^A_{t+1}(s_t, a_t) = Q^A_t(s_t, a_t) + \alpha_t(s_t, a_t)\left(r_t + \gamma Q^B_t\left(s_{t+1}, \arg\max_a Q^A_t(s_{t+1}, a)\right) - Q^A_t(s_t, a_t)\right), \text{ and}$$

$$Q^B_{t+1}(s_t, a_t) = Q^B_t(s_t, a_t) + \alpha_t(s_t, a_t)\left(r_t + \gamma Q^A_t\left(s_{t+1}, \arg\max_a Q^B_t(s_{t+1}, a)\right) - Q^B_t(s_t, a_t)\right).$$

Literature Review (Cont.)

- Deep Deterministic Policy Gradient (DDPG) [Lillicrap, et al., 2015]

$$Q^{\mu}(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}\left[r(s_t, a_t) + \gamma Q^{\mu}(s_{t+1}, \mu(s_{t+1}))\right]$$

- Clipped Double Q-Learning [Fujimoto et. al., 2018]

$$y_1 = r + \gamma \min_{i=1,2} Q_{\theta_i'}(s', \pi_{\phi_1}(s'))$$

- Imitation Learning: This method used when an expert can demonstrate the desired behavior, **imitating** the expert, however, we do not have access to the reward function.
  - Behavior Cloning: Uses supervised learning from demonstration data to estimate the expert policy; however, suffers distribution shift [Ross, et al., 2010]

$$\hat{\pi} = \arg\min_{\pi \in \Pi} \mathbb{E}_{s \sim d_{\pi}}\left[\ell(s, \pi)\right]$$

  - Inverse RL: Uses the estimated reward function from the demonstration to generate the policy. [Ng, et. al, 2000]

Sparse Rewards

- An ideal reward system takes into account
  - Final Goal
  - Policy Optimization Guidance
- The conventional algorithm takes in account that we have a domain expert who/which can design the rewards well enough to make it dense.
  - Uses a lot of resources
  - Not applicable to experiments that the algorithms have not enough knowledge about the admissible behaviors

Learning Online with Guidance Offline (Baseline)

- Learning Online with Guidance Offline (LOGO) [Rengarajan et al., 2022]
  - Classic TRPO one step policy improvement and then we guide the policy in the direction of the behavioral policy given by the domain expert, this is governed by the KL divergence between our policy and the expert policy.
  - **Policy Improvement**, where delta is the trust region

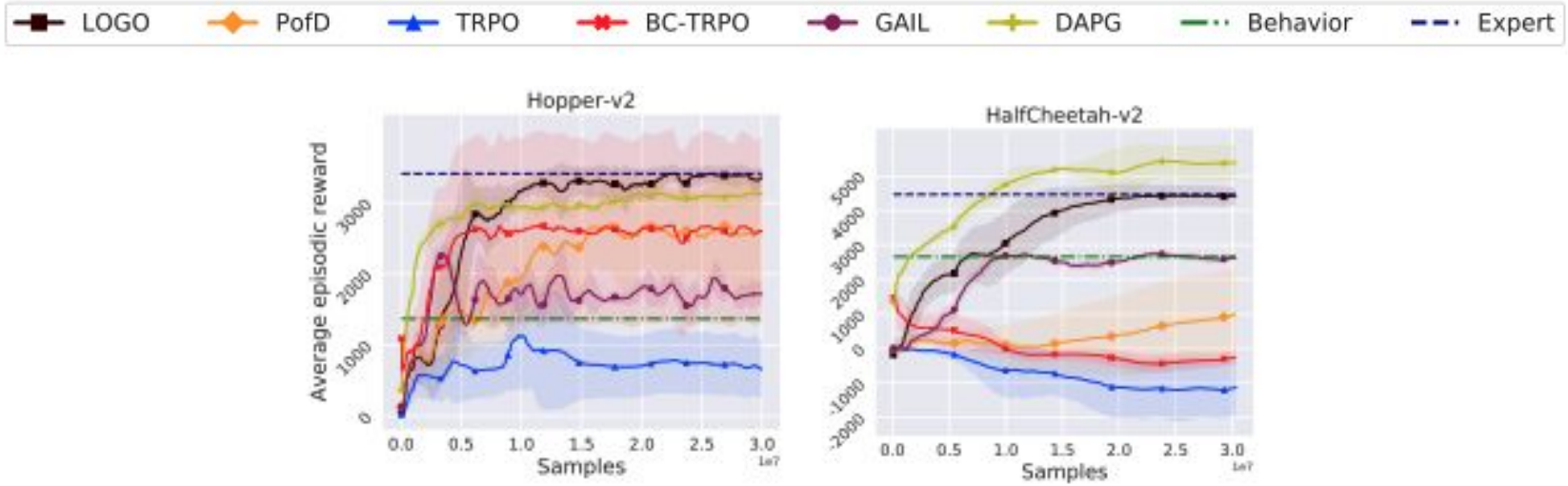
$$\pi_{k+1/2} = \arg\max_{\pi} \; \mathbb{E}_{s\sim d^{\pi_k}, a\sim\pi}\left[A_R^{\pi_k}(s,a)\right] \quad \text{s.t.} \quad D_{\mathrm{KL}}^{\pi_k}(\pi,\pi_k) \leq \delta.$$

  - **Policy Guidance,** her is the delta gradually decays as the agent has explored in the direction of the behavior policy

$$\pi_{k+1} = \arg\min_{\pi} \; D_{\mathrm{KL}}^{\pi}(\pi,\pi_{\mathrm{b}}) \quad \text{s.t.} \quad D_{\mathrm{KL}}^{\max}(\pi,\pi_{k+1/2}) \leq \delta_k$$

*Read the paper for the complete algorithms, which includes a practical implementation that utilizes the TRPO codebase via implementing a surrogate function to approximate the advantages.

Results of LOGO

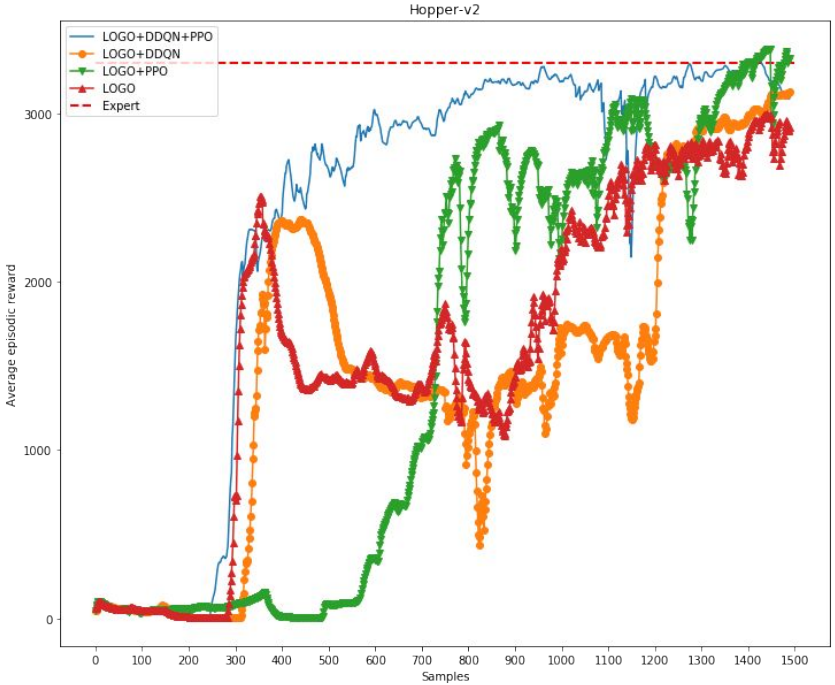


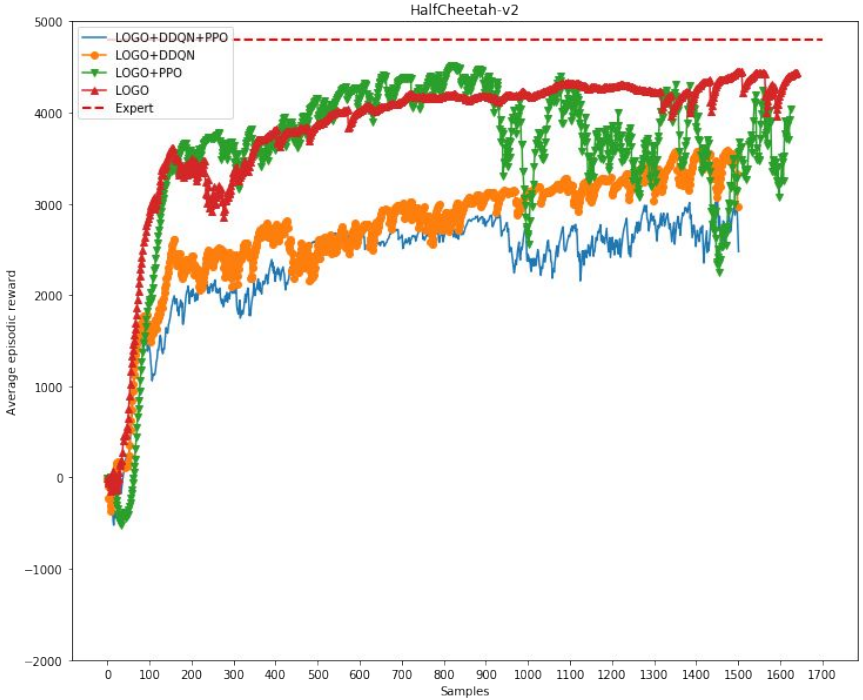
Our Approach

As seen from the previous slide: LOGO is essentially a two step TRPO. We can make the following two modifications to improve the LOGO performance:

- Clipped Double Q-Learning as the Critic [Fujimoto et. al., 2018]
- PPO to improve TRPO performance [Schulman et al. 2017]

Experiments and Results





Future Directions

- Explore Off-Policy Approaches
  - Hindsight Experience Replay [Andrychowicz, et al., 2018]
- No demonstrations available
  - Bayesian Optimization Based Algorithms
    - Sparse Bayesian Optimization [Liu, et al., 2022]
  - Monte-Carlo Tree Search Algorithms
    - Monte-Carlo Tree Search as Regularized Policy Optimization [Grill, et al.,2020]

References

- A. Y. Ng, S. J. Russell, ‘Algorithms for Inverse Reinforcement Learning’, στο *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, σσ. 663–670.
- D. Rengarajan, G. Vaidya, A. Sarvesh, D. Kalathil, S. Shakkottai, ‘Reinforcement Learning with Sparse Rewards using Guidance from Offline Demonstration’. arXiv, 2022.
- J.-B. Grill., ‘Monte-Carlo Tree Search as Regularized Policy Optimization’. arXiv, 2020.
- J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, ‘Proximal Policy Optimization Algorithms’. arXiv, 2017.
- H. van Hasselt, A. Guez, D. Silver, ‘Deep Reinforcement Learning with Double Q-learning’. arXiv, 2015.
- M. Andrychowicz, ‘Hindsight Experience Replay’. arXiv, 2017.
- S. Fujimoto, H. van Hoof, D. Meger, ‘Addressing Function Approximation Error in Actor-Critic Methods’. arXiv, 2018.
- S. Liu, Q. Feng, D. Eriksson, B. Letham, και E. Bakshy, ‘Sparse Bayesian Optimization’. arXiv, 2022.
- S. Ross, G. J. Gordon, και J. A. Bagnell, ‘A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning’. arXiv, 2010.
- T. P. Lillicrap, ‘Continuous control with deep reinforcement learning’. arXiv, 2015.
- Y. Li, ‘Deep Reinforcement Learning: An Overview’. arXiv, 2017.