Recitation 8:

Quiz 3 Review

Overview:

Main Topics:

- All papers not marked “optional” since and including 09/27
- PPO, TRPO
- PETS
- MBPO
- DDPG
- AlphaGo, AlphaGoZero
- Evolutionary Strategies
- BC, DAGGER
- GAIL

Quiz Date: Monday 11/08 During Class

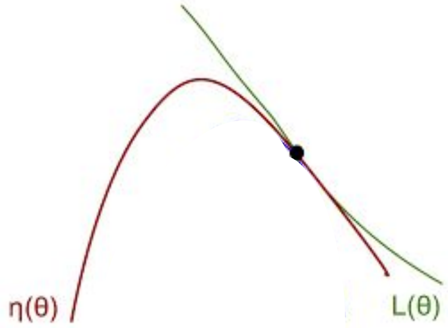
Trust Region Policy Optimization (TRPO) Motivation

Goal: $$\max_{\theta} \eta(\theta) = \mathbb{E}_{s_0, a_0, \ldots}\left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t\right)\right] = \mathbb{E}_{s \sim \rho_{\pi_\theta}}\left[\sum_{t=0}^{\infty} R(s)\right]$$

Exact Update: $$\eta\left(\theta_{\text{new}}\right) = \eta(\theta_{\text{old}}) + \sum_{s} \rho_{\pi_{\theta_{\text{new}}}}(s) \sum_{a} \pi_{\theta_{\text{new}}}(a \mid s) A_{\theta_{\text{old}}}(s, a)$$

Approximation:

$$L_{\theta_{\text{old}}}\left(\theta_{\text{new}}\right) = \eta(\theta_{\text{old}}) + \sum_{s} \rho_{\pi_{\theta_{\text{old}}}}(s) \sum_{a} \pi_{\theta_{\text{new}}}(a \mid s) A_{\theta_{\text{old}}}(s, a)$$

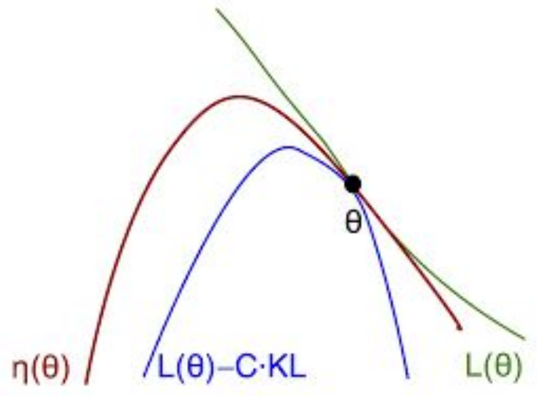


TRPO: monotonic improvement Theorem

We want to construct a lower bound on $\eta(\theta)$

Theorem: $\eta(\theta_{\text{new}}) \geq L_{\theta_{\text{old}}}(\theta_{\text{new}}) - C D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta_{\text{new}})$

where $C = \frac{4\epsilon\gamma}{(1-\gamma)^2}, \epsilon = \max_{s,a} |A_{\theta_{\text{old}}}(s,a)|$



TRPO: In practice

The monotonic improvement Theorem proposes too small steps in practice.

Instead: $\max_{\theta} L_{\theta_{\text{old}}}(\theta)$

subject to $D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \leq \delta$

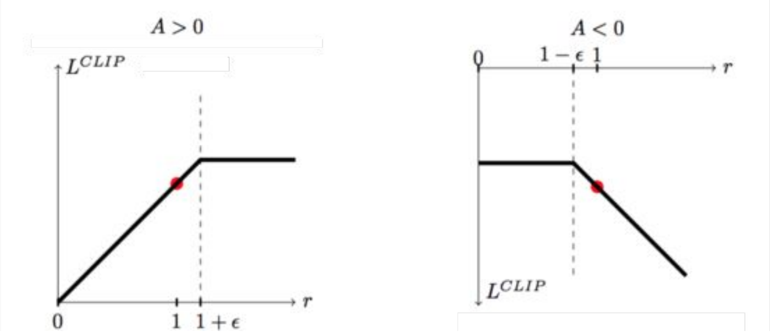
To compute this, TRPO uses the conjugate gradient method, requiring computing the Fisher Information Matrix (FIM), i.e. the the Hessian of the KL divergence.

Proximal Policy Optimization (PPO)

Can we simplify the KL constraint, so we don't have to compute the FIM?

Let the ratio $r_t(\theta) = \pi_\theta\left(a_t \mid s_t\right) / \pi_{\theta_k}\left(a_t \mid s_t\right)$

Define: $\mathcal{L}_{\theta_{\text{old}}}^{CLIP}(\theta) = \underset{\tau \sim \pi_{\text{old}}}{\mathrm{E}}\left[\sum_{t=0}^{T}\left[\min\left(r_t(\theta)\hat{A}_t^{\pi_{\text{old}}}, \operatorname{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t^{\pi_{\text{old}}}\right)\right]\right]$



Actor-Critic methods (Quick refresher)

Objective: $\max_{\theta} .\mathbb{E}_{\tau\sim P_\theta(\tau)}[R(\tau)]$

Gradient update: $\mathbb{E}_{s\sim d^{\pi}\theta(s),a\sim\pi_\theta(a|s)}\nabla_\theta \log \pi_\theta(a \mid s)[A(s,a;\phi))]$

Deep Deterministic Policy Gradient (DDPG)

What if we instead represent our policy as a deterministic function.

$$\pi : S \mapsto A$$

If our learned Q-function is differentiable, we can directly optimize our policy to maximize the Q-function.

$$\max_{\theta} \mathbb{E}_{\pi_\theta} \left[Q\left(s, \pi_\theta(s)\right) \right]$$

Gradient update:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q\left(s, a \mid \theta^Q\right)\Big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu\left(s \mid \theta^\mu\right)\Big|_{s_i}$$

To explore, add Gaussian noise during training.

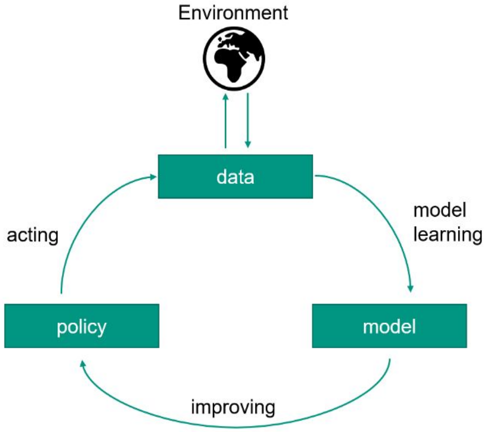
Soft Actor-Critic (SAC)

Increasing the entropy of our policy improves exploration and makes it less likely to get stuck in a local minimum.

SAC adds a entropy reward to encourage higher entropy policies.

$$\max_{\pi} E_{\pi}\left[\sum_{t} \gamma^{t}\left(r\left(s_{t}, a_{t}\right)+\mathcal{H}_{\pi}\left[a_{t} \mid s_{t}\right]\right)\right]$$

Model-based RL (MBRL)

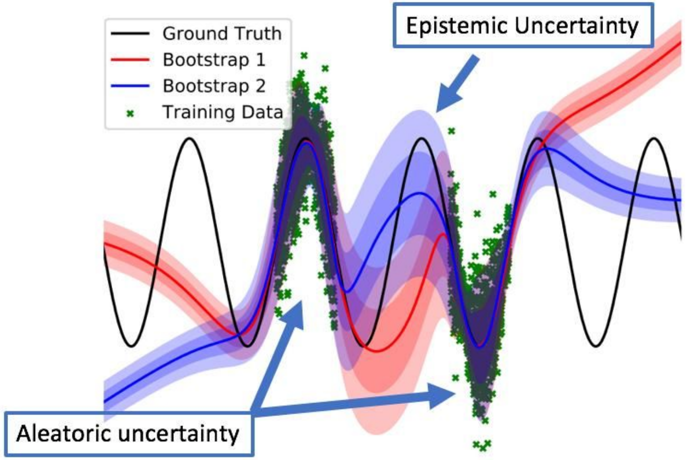


becominghuman.ai

Main challenge for MBRL: how to deal with model errors?

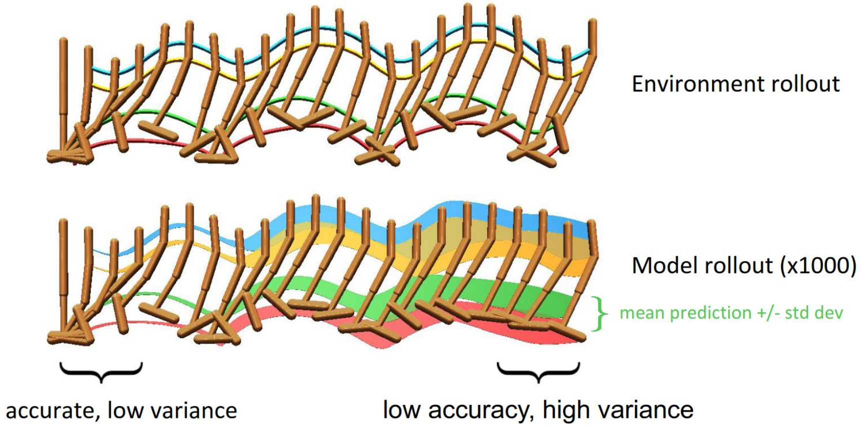
Probabilistic Ensembles with Trajectory Sampling (PETS)

- Uses probabilistic models to estimate Aleatoric uncertainty
- Uses an ensemble of such models to estimate Epistemic uncertainty
- Uses Model Predictive Control (MPC) to compute a new policy at every step:
    - Use particle-based sampling method to estimate returns for a given policy
    - Use CEM to optimize the policy



Model Based Policy Optimization (MBPO)

- Longer trajectories result in larger errors
- By simply decreasing the trajectory length, we can increase accuracy
    - Note: this only works on domains with dense reward functions
- MBPO uses SAC to optimize the policy

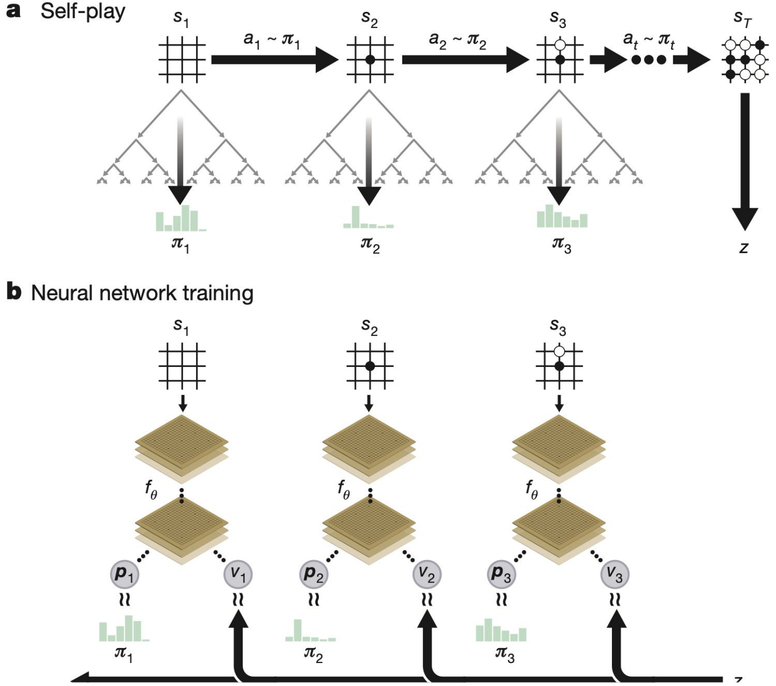


AlphaGo

- Go, massive state-action space
- Use combined policy, value network to prune MCTS tree depth and breadth
- 19x19 board passed through CNN for feature extraction
- Start with SL p_sigma, p_pi from expert moves
- Then train p_ro with RL to optimize game outcomes in self-play
    - Need to re-adjust objective to winning game
- Finally train v_theta that predicts winners of self-play games

AlphaGo Zero

- Combine policy/value networks into single architecture
- No expert knowledge needed
- Uses MCTS as a policy improvement operator
- Each time step is used to update model parameters using (s_t, pi, z_t) tuples



Evolutionary Strategies

- Permute → Test mutations → Select the best for the next round → Repeat
- "Black-box" optimization
  - Directly search over policy parameter space
  - No gradient information
  - No reward information
  - No state-structure information
  - May get stuck at local optima
- CEM (Cross Entropy Method)
  - Sample using a multivariate gaussian w/ diagonal covariance (independent dimensions)
  - Update mean and variance using elites
  - Great low-dimensional performance
- CMA-ES (Covariance Matrix Adaptation Evolutionary Strategy)
  - Sample using a multivariate gaussian w/ full covariance (correlated dimensions)
  - Update mean and variance using elites

Behavioral Cloning (BC), DAGGER

- Methods for imitation learning (making a novice act like an expert)
- BC:
  - Use the expert to generate and label random training data
  - Use supervised learning to train the novice
  - Problem: Distribution Shift between Test
- Solution: Data Aggregation (DAGGER)
  - “Fuse” the expert and novice distributions by using the expert to relabel the novice’s own attempts
  - Keep learning from your past knowledge with aggregation
    - No “forgetting”

Generative Adversarial Imitation Learning (GAIL)

- Learn without a reward signal or interaction with an expert
  - Start only with a set of trajectories from the expert, no more querying allowed
- Use a Discriminator network as a cost function to train a parameterized policy
  - D(s,a) → [0,1]
  - Policy loss: log(D_updated(s,a)), use TRPO for update step
- Outperforms BC

Ultimate objective is to find a saddle point of:

$$\mathbb{E}_{\pi}[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] - \lambda H(\pi)$$

Some Questions...

When does GAIL outperform other IL methods (i.e., BC & DAGGER) and why ?

Some Questions...

When does GAIL outperform other IL methods (i.e., BC & DAGGER) and why ?

GAIL works better when the expert trajectory distribution is multi-modal, because minimizing KL divergence is only suited to approximate uni-modal expert distribution.

Some Questions...

What is maximization bias?

Some Questions...

What is maximization bias?

Overestimation of a value function due to the fact that the same function which is being optimized is used to evaluate its own performance. (Happens on q-value bootstrapping step)

Some Questions...

Describe one of the drawbacks of DDPG.

Some Questions...

Describe one of the drawbacks of DDPG.

DDPG can be unstable and heavily reliant on finding the correct hyper parameters for the current task. This is caused by the algorithm continuously over estimating the Q values of the critic (value) network.