Trust Region Policy Optimization

TINGWU WANG

MACHINE LEARNING GROUP,

UNIVERSITY OF TORONTO

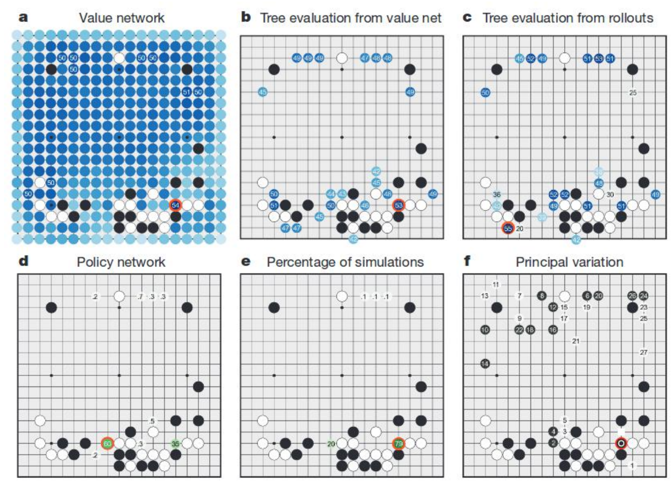
Contents

Introduction

1. **Introduction**
   1. **Problem Domain: Locomotion**
   2. **Related Work**
2. TRPO Step-by-step
   1. The Preliminaries
   2. Find the Lower-Bound in General Stochastic policies
   3. Optimization of the Parameterized Policies
   4. From Math to Practical Algorithm
   5. Tricks and Efficiency
   6. Summary
3. Misc
   1. Results and Problems of TRPO

Problem Domain: Locomotion

1. The two action domains in reinforcement learning:
    1. Discrete action space
        1. Only several actions are available (up, down, left, right)
        2. Q-value based methods (DQN [1], or DQN + MCTS [2])



Problem Domain: Locomotion

1. The two action domains in reinforcement learning:
    1. Discrete action space
    2. Continuous action space
        1. One of the most interesting problems: locomotion
        2. MuJuCo: A physics engine for model-based control [3]
        3. TRPO [4] (today's focus)
            1. One of the most important baselines in model-free continuous control problem [5]
            2. It works for discrete action space too

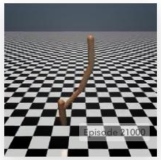
Walker2d-v1
Make a 2D robot walk.

Ant-v1
Make a 3D four-legged robot walk.

Humanoid-v1
Make a 3D two-legged robot walk.

HalfCheetah-v1
Make a 2D cheetah robot run.

Swimmer-v1
Make a 2D robot swim.



Hopper-v1
Make a 2D robot hop.

Problem Domain: Locomotion

1. The two action domains in reinforcement learning:
    1. Discrete action space
    2. Continuous action space
    3. Difference between Discrete & Continuous
        1. Raw-pixel Input
            1. Control versus perception
        2. Dynamical Model
            1. Game dynamics versus physical models
        3. Reward Shaping
            1. Zero-one reward versus continous reward at evert time step

Related Work

1. REINFORCE algorithm [6]

$$\widehat{\nabla_\theta \eta(\pi_\theta)} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=0}^{T} \nabla_\theta \log \pi(a_t^i | s_t^i; \theta)(R_t^i - b_t^i)$$

2. Deep Deterministic Policy Gradient [7]

$$\widehat{\nabla_\theta \eta(\mu_\theta)} = \sum_{i=1}^{B} \nabla_a Q_\phi(s_i, a)|_{a=\mu_\theta(s_i)} \nabla_\theta \mu_\theta(s_i)$$

3. TNPG method [8]
   1. Very similar to the TRPO
   2. TRPO uses a fixed KL divergence rather than a fixed penalty coefficient
   3. Similar performance according to Duan [9]

TROO Step-by-step

1. Introduction
    1. Problem Domain: Locomotion
    2. Related Work
2. **TRPO Step-by-step**
    1. **The Preliminaries**
    2. **Find the Lower-Bound in General Stochastic policies**
    3. **Optimization of the Parameterized Policies**
    4. **From Math to Practical Algorithm**
    5. **Tricks and Efficiency**
    6. **Summary**
3. Misc
    1. Results and Problems of TRPO

The Preliminaries

1. The objective function to optimize

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t) \right], \text{ where}$$

$$s_0 \sim \rho_0(s_0),\ a_t \sim \pi(a_t|s_t),\ s_{t+1} \sim P(s_{t+1}|s_t, a_t)$$

2. Can we expresses the expected return of another policy in terms of the advantage over the original policy?

Yes, orginally proven in [8] (see whiteboard 1). It shows that a guaranteed increase in the performance is possible.

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t) \right] = \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a).$$

The Preliminaries

3. Can we remove the dependency of discounted visitation frequencies under the new policy?
   1. The local approximation

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s,a).$$

$$L_{\pi_{\theta_0}}(\pi_{\theta_0}) = \eta(\pi_{\theta_0}),$$
$$\nabla_\theta L_{\pi_{\theta_0}}(\pi_\theta)\big|_{\theta=\theta_0} = \nabla_\theta \eta(\pi_\theta)\big|_{\theta=\theta_0}.$$

   2. The lower bound from conservative policy iteration [8]

$$\pi_{\text{new}}(a|s) = (1-\alpha)\pi_{\text{old}}(a|s) + \alpha\pi'(a|s).$$
$$\eta(\pi_{\text{new}}) \geq L_{\pi_{\text{old}}}(\pi_{\text{new}}) - \frac{2\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$
$$\text{where } \epsilon = \max_s \left| \mathbb{E}_{a \sim \pi'(a|s)} \left[A_\pi(s,a) \right] \right|.$$

Find the Lower-Bound in General Stochastic policies

1. Can we move the be extended to general stochastic policies, rather than just mixture polices? (see whiteboard)

$$\pi_{\mathrm{new}}(a|s) = (1-\alpha)\pi_{\mathrm{old}}(a|s) + \alpha\pi'(a|s).$$

$$\eta(\pi_{\mathrm{new}}) \geq L_{\pi_{\mathrm{old}}}(\pi_{\mathrm{new}}) - \frac{2\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$

$$\text{where } \epsilon = \max_s \left| \mathbb{E}_{a\sim\pi'(a|s)}\left[A_\pi(s,a)\right] \right|.$$

Theorem 1. *Let* $\alpha = D_{\mathrm{TV}}^{\max}(\pi_{\mathrm{old}}, \pi_{\mathrm{new}})$. *Then the following bound holds:*

$$\eta(\pi_{\mathrm{new}}) \geq L_{\pi_{\mathrm{old}}}(\pi_{\mathrm{new}}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$

$$\text{where } \epsilon = \max_{s,a} |A_\pi(s,a)| \qquad (8)$$

2. Maybe even make the equation simpler?

$$\eta(\tilde{\pi}) \geq L_\pi(\tilde{\pi}) - C D_{\mathrm{KL}}^{\max}(\pi, \tilde{\pi}),$$

$$\text{where } C = \frac{4\epsilon\gamma}{(1-\gamma)^2}.$$

(later we make it even easier by approximate the maximum of KL using the average of KL)

Find the Lower-Bound in General Stochastic policies

3. Now what's the objective function we are trying to maximize?

let $M_i(\pi) = L_{\pi_i}(\pi) - CD_{\mathrm{KL}}^{\max}(\pi_i, \pi)$. Then

$$\eta(\pi_{i+1}) \geq M_i(\pi_{i+1}) \text{ by Equation (9)}$$
$$\eta(\pi_i) = M_i(\pi_i), \text{ therefore,}$$
$$\eta(\pi_{i+1}) - \eta(\pi_i) \geq M_i(\pi_{i+1}) - M(\pi_i).$$

$$\eta(\tilde{\pi}) \geq L_\pi(\tilde{\pi}) - CD_{\mathrm{KL}}^{\max}(\pi, \tilde{\pi}),$$
$$\text{where } C = \frac{4\epsilon\gamma}{(1-\gamma)^2}.$$

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s,a).$$

$$L_{\pi_{\theta_0}}(\pi_{\theta_0}) = \eta(\pi_{\theta_0}),$$
$$\nabla_\theta L_{\pi_{\theta_0}}(\pi_\theta)\big|_{\theta=\theta_0} = \nabla_\theta \eta(\pi_\theta)\big|_{\theta=\theta_0}.$$

Guaranteed Improvement! (minorization-maximization algorithm)

Optimization of the Parameterized Policies

1. In practice, if we used the penalty coefficient C recommended by the theory above, the step sizes would be very small.

$$\underset{\theta}{\text{maximize}} \left[L_{\theta_{\text{old}}}(\theta) - C D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \right]$$

2. One way to take larger steps in a robust way is to use a constraint on the KL divergence between the new policy and the old policy, i.e., a trust region constraint
    1. Use the average KL instead of the maximum of the KL (heuristic approximation)

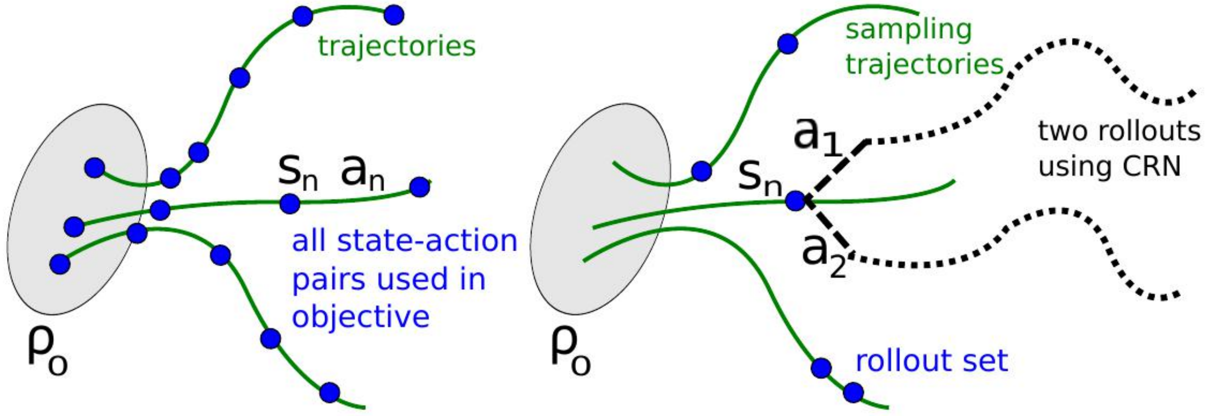
$$\underset{\theta}{\text{maximize}}\ L_{\theta_{\text{old}}}(\theta) \quad \text{subject to } D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \leq \delta.$$

$$\underset{\theta}{\text{maximize}}\ L_{\theta_{\text{old}}}(\theta) \quad \text{subject to } \overline{D}_{\text{KL}}^{\rho_{\theta_{\text{old}}}}(\theta_{\text{old}}, \theta) \leq \delta.$$

From Math to Practical Algorithm

1. Sample-Based Estimation of the Objective and Constraint

$$\underset{\theta}{\text{maximize}}\, \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}},a\sim q}\left[\frac{\pi_\theta(a|s)}{q(a|s)}Q_{\theta_{\text{old}}}(s,a)\right]$$
$$\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}}\left[D_{\text{KL}}(\pi_{\theta_{\text{old}}}(\cdot|s)\,\|\,\pi_\theta(\cdot|s))\right]\leq\delta.$$



Tricks and Efficiency

1. Search for the next parameter

   maximize $L(\theta)$ subject to $\overline{D}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta) \leq \delta$.

   1. Compute a search direction, using a linear approximation to objective and quadratic approximation to the constraint

      $$Ax = g$$

      $$\overline{D}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta) \approx \frac{1}{2}(\theta - \theta_{\mathrm{old}})^T A(\theta - \theta_{\mathrm{old}}) \quad A_{ij} = \frac{\partial}{\partial \theta_i} \frac{\partial}{\partial \theta_j} \overline{D}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta)$$

   2. Use conjugate gradient algorithm to solve $Ax = b$
   3. Get the maximal step length and decay exponentially

      $$\delta = \overline{D}_{\mathrm{KL}} \approx \frac{1}{2}(\beta s)^T A(\beta s) = \frac{1}{2}\beta^2 s^T A s$$

      $$\beta = \sqrt{2\delta / s^T A s}.$$

      $$L_{\theta_{\mathrm{old}}}(\theta) - \mathcal{X}[\overline{D}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta) \leq \delta]$$

Summary

1. The original objective

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t) \right], \text{ where}$$

$$s_0 \sim \rho_0(s_0),\ a_t \sim \pi(a_t|s_t),\ s_{t+1} \sim P(s_{t+1}|s_t, a_t)$$

2. The objective of another policy in terms of the advantage over the original policy

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t) \right] = \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a).$$

3. Remove the dependency on the trajectories of new policy.

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a).$$

$$L_{\pi_{\theta_0}}(\pi_{\theta_0}) = \eta(\pi_{\theta_0}),$$

$$\nabla_\theta L_{\pi_{\theta_0}}(\pi_\theta)\big|_{\theta=\theta_0} = \nabla_\theta \eta(\pi_\theta)\big|_{\theta=\theta_0}.$$

Summary

4. Find the lower-bound that guarantees the improvement

let $M_i(\pi) = L_{\pi_i}(\pi) - CD_{\mathrm{KL}}^{\max}(\pi_i, \pi)$. Then

$$\eta(\pi_{i+1}) \geq M_i(\pi_{i+1}) \text{ by Equation (9)}$$
$$\eta(\pi_i) = M_i(\pi_i), \text{ therefore,}$$
$$\eta(\pi_{i+1}) - \eta(\pi_i) \geq M_i(\pi_{i+1}) - M(\pi_i).$$

5. Sample-based estimation

$$\underset{\theta}{\text{maximize}}\, \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}, a\sim q}\left[\frac{\pi_\theta(a|s)}{q(a|s)} Q_{\theta_{\text{old}}}(s,a)\right]$$
$$\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}}\left[D_{\mathrm{KL}}(\pi_{\theta_{\text{old}}}(\cdot|s) \,\|\, \pi_\theta(\cdot|s))\right] \leq \delta.$$

6. Using line-search (Approximation, Fisher matrix, Conjugate gradient)

$$\delta = \overline{D}_{\mathrm{KL}} \approx \tfrac{1}{2}(\beta s)^T A(\beta s) = \tfrac{1}{2}\beta^2 s^T A s$$
$$\beta = \sqrt{2\delta / s^T A s}$$
$$L_{\theta_{\text{old}}}(\theta) - \mathcal{X}[\overline{D}_{\mathrm{KL}}(\theta_{\text{old}}, \theta) \leq \delta]$$

Misc

Results and Problems of TRPO

1. Results
   1. One of the most successful baselines in locomotion

2. Problems
   1. Sample inefficiency
   2. Unable to scale to big network

Task	Random	REINFORCE	TNPG	RWR	REPS	TRPO	CEM	CMA-ES	DDPG
Cart-Pole Balancing	77.1 ± 0.0	4693.7 ± 14.0	**3986.4 ± 748.9**	**4861.5 ± 12.3**	565.6 ± 137.6	**4869.8 ± 37.6**	4815.4 ± 4.8	2440.4 ± 568.3	4634.4 ± 87.8
Inverted Pendulum*	−153.4 ± 0.2	13.4 ± 18.0	**209.7 ± 55.5**	84.7 ± 13.8	−113.3 ± 4.6	**247.2 ± 76.1**	38.2 ± 25.7	−40.1 ± 5.7	40.0 ± 244.6
Mountain Car	−415.4 ± 0.0	−67.1 ± 1.0	**-66.5 ± 4.5**	−79.4 ± 1.1	−275.6 ± 166.3	**-61.7 ± 0.9**	−66.0 ± 2.4	−85.0 ± 7.7	−288.4 ± 170.3
Acrobot	−1904.5 ± 1.0	−508.1 ± 91.0	−395.8 ± 121.2	−352.7 ± 35.9	−1001.5 ± 10.8	−326.0 ± 24.4	−436.8 ± 14.7	−785.6 ± 13.1	**-223.6 ± 5.8**
Double Inverted Pendulum*	149.7 ± 0.1	4116.5 ± 65.2	**4455.4 ± 37.6**	3614.8 ± 368.1	446.7 ± 114.8	**4412.4 ± 50.4**	2566.2 ± 178.9	1576.1 ± 51.3	2863.4 ± 154.0
Swimmer*	−1.7 ± 0.1	92.3 ± 0.1	**96.0 ± 0.2**	60.7 ± 5.5	3.8 ± 3.3	**96.0 ± 0.2**	68.8 ± 2.4	64.9 ± 1.4	85.8 ± 1.8
Hopper	8.4 ± 0.0	714.0 ± 29.3	**1155.1 ± 57.9**	553.2 ± 71.0	86.7 ± 17.6	**1183.3 ± 150.0**	63.1 ± 7.8	20.3 ± 14.3	267.1 ± 43.5
2D Walker	−1.7 ± 0.0	506.5 ± 78.8	**1382.6 ± 108.2**	136.0 ± 15.9	−37.0 ± 38.1	**1353.8 ± 85.0**	84.5 ± 19.2	77.1 ± 24.3	318.4 ± 181.6
Half-Cheetah	−90.8 ± 0.3	1183.1 ± 69.2	**1729.5 ± 184.6**	376.1 ± 28.2	34.5 ± 38.0	**1914.0 ± 120.1**	330.4 ± 274.8	441.3 ± 107.6	**2148.6 ± 702.7**
Ant*	13.4 ± 0.7	548.3 ± 55.5	**706.0 ± 127.7**	37.6 ± 3.1	39.0 ± 9.8	**730.2 ± 61.3**	49.2 ± 5.9	17.8 ± 15.5	326.2 ± 20.8
Simple Humanoid	41.5 ± 0.2	128.1 ± 34.0	**255.0 ± 24.5**	93.3 ± 17.4	28.3 ± 4.7	**269.7 ± 40.3**	60.6 ± 12.9	28.7 ± 3.9	99.4 ± 28.1
Full Humanoid	13.2 ± 0.1	262.2 ± 10.5	**288.4 ± 25.2**	46.7 ± 5.6	41.7 ± 6.1	**287.0 ± 23.4**	36.9 ± 2.9	N/A ± N/A	119.0 ± 31.2
Cart-Pole Balancing (LS)*	77.1 ± 0.0	420.9 ± 265.5	**945.1 ± 27.8**	68.9 ± 1.5	898.1 ± 22.1	**960.2 ± 46.0**	227.0 ± 223.0	68.0 ± 1.6	
Inverted Pendulum (LS)	−122.1 ± 0.1	−13.4 ± 3.2	**0.7 ± 6.1**	−107.4 ± 0.2	−87.2 ± 8.0	**4.5 ± 4.1**	−81.2 ± 33.2	−62.4 ± 3.4	
Mountain Car (LS)	−83.0 ± 0.0	−81.2 ± 0.6	**-65.7** ± 9.0	−81.7 ± 0.1	−82.6 ± 0.4	**-64.2 ± 9.5**	**-68.9 ± 1.3**	**-73.2 ± 0.6**	
Acrobot (LS)*	−393.2 ± 0.0	−128.9 ± 11.6	**-84.6 ± 2.9**	−235.9 ± 5.3	−379.5 ± 1.4	**-83.3 ± 9.9**	−149.5 ± 15.3	−159.9 ± 7.5	

References

[1] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529.

[2] Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." Nature 529.7587 (2016): 484.

[3] Erez, Tom, Yuval Tassa, and Emanuel Todorov. "Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX." Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE, 2015.

[4] Schulman, John, et al. "Trust region policy optimization." Proceedings of the 32nd International Conference on Machine Learning (ICML-15). 2015.

[5] Duan, Yan, et al. "Benchmarking deep reinforcement learning for continuous control." Proceedings of the 33rd International Conference on Machine Learning (ICML). 2016.

[6] Williams, Ronald J. "Simple statistical gradient-following algorithms for connectionist reinforcement learning." Machine learning 8.3-4 (1992): 229-256.

[7] Lillicrap, Timothy P., et al. "Continuous control with deep reinforcement learning." arXiv preprint arXiv:1509.02971 (2015).

[8] Kakade, Sham. "A natural policy gradient." Advances in neural information processing systems 2 (2002): 1531-1538.

Q&A

Thanks for listening ;P