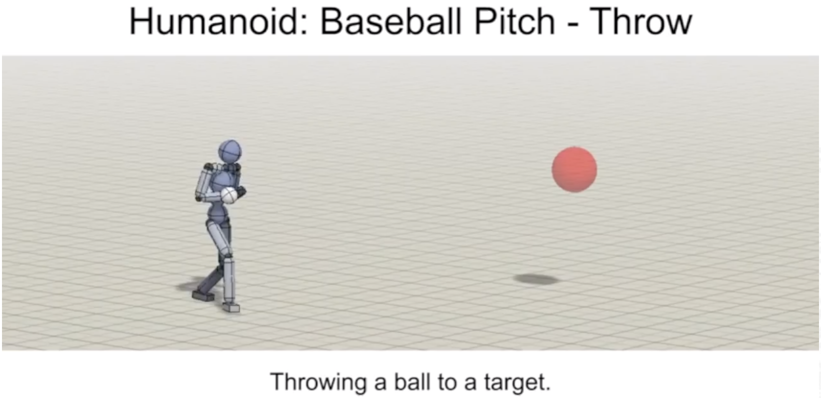
Soft Actor-Critic: Deep Reinforcement Learning for Robotics

Finn Rietz

University of Hamburg
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics
Technical Aspects of Multimodal Systems

13. January 2020

Creative policy example



Taken from [1]

Outline

1. Motivation and reinforcement learning (RL) basics
2. Challenges in deep reinforcement learnign (DRL) with robotics
3. Soft actor-critic algorithm
4. Results and Discussion
5. Conclusion

Motivation

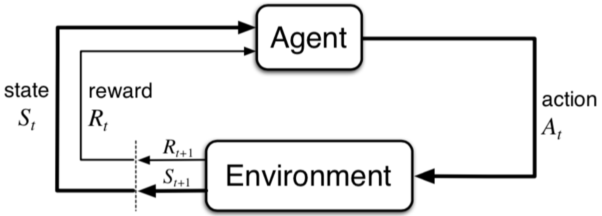
Potential of RL:

- Automatic learning of robotic tasks, directly from sensory input

Promising results:

- Superhuman performance on Atari games [2]
- AlphaGoZero becoming the greatest Go player [3]
- AlphaStart becoming better than 99.8% of all Star Craft 2 players [4]
- Real-world, simple robotic manipulation tasks (numerous limitations) [5]

Basics



Markov Decision Process. Figure taken from [6]

RL in a nutshell:

- Learning to map actions to situations
- Trial-and-error search
- Maximize numerical reward

- Reward r_t: Skalar
- State function $\mathbf{s}_t \in S$: Vector of observations
- Action function $\mathbf{a}_t \in A$: Vector of actions
- Policy π: Mapping function from states to actions
- Action-Value function $Q_\pi(\mathbf{s}_t, \mathbf{a}_t)$: Expected reward for state-action pair

Putting the deep in RL:

- How to deal with continuous spaces?
- Approximate (state and action) function
- Approximator has fewer, limited number of parameters

On-policy versus off-policy learning

On-policy learning:

- Only one policy
- Exploitation versus exploration dilemma
- Optimize same policy that collects data
- Very data hungry

Off-policy learning:

- Employs multiple policies
- One collects data, other becomes final policy
- We can save and reuse past experiences
- More suitable for robotics

Model-based versus model-free methods

Model-based methods:

- Learn model of the environment
- Chose actions by planning on learned model
- "Think then act"
- Statistically efficient, but model often too complex to learn

Model-free methods:

- Directly learn Q-function by sampling from environment
- No planning possible
- Can produce same optimal policy as model-based methods
- More suitable for robotics

Progress

Data inefficiency

RL algorithms are notoriously data-hungry:

- Not a big problem in simulated settings
- Impractical amounts of training time in real-world
- Wear-and-tear on robot must be minimized
- Need for statistically efficient methods

Off-policy methods better suited, due to higher sample-efficiency

Safe exploration

RL is trial-and-error search:

- Again no problem in simulation
- Randomly applying force to motors of an expansive robot is problematic
- Could lead to destruction of robot
- Need for safety measures during exploration

Possible solutions: Limit maximum allowed velocity per joint, position limits for joints [7]

Sparse rewards

Classic reward is binary measure:

- Robot might never complete complex tasks, thus never observes reward
- No variance in reward function, no learning possible
- Need for manually designed reward function, *reward engineering*
- Need for designated state representation, against the principal of RL
- Not trivial problem, manually designed reward function often exploited in an unforeseen manner

Reality Gap

Why not train in simulation?

- Simulations are still imperfect
- Many (small) dynamics of the environment remain uncaptured
- Policy will likely not generalize to real world
- Recent research field (automatic domain randomization)

Training in simulation more attractive, but often policy not directly applicable in the real world

Progress

Soft actor-critic algorithm

Soft actor-critic by Haarnoja et al:

- Original version early 2018: Temperature hyperparameter [8]
- Refined version late 2018: Workaround for critical hyperparameter [9]
- Developed in cooperation by UC Berkeley & Google Brain

- Off-policy, model-free, actor-critic method
- Key-idea: Exploit entropy of policy
- "Succeed at task while acting as random as possible" [9]

Soft actor-critic algorithm

Classical reinforcement learning objective:

- $\sum_t \mathbb{E}(\mathbf{s}_t, \mathbf{a}_t)_{\sim\rho_\pi}[r(\mathbf{s}_t, \mathbf{a}_t)]$
- Find $\pi(\mathbf{a}_t|\mathbf{s}_t)$ maximizing sum of reward

SAC objective:

- $\pi^* = \underset{\pi}{\operatorname{argmax}} \sum_t \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t)\sim\rho_\pi}[r(\mathbf{s}_t, \mathbf{a}_t) + \alpha\mathcal{H}(\pi(\cdot|\mathbf{s}_t))]$
- Augment classical objective with entropy regularization $\mathcal{H}$
- Problematic hyperparameter α

- Instead treat entropy as constraint, automatically update during learning

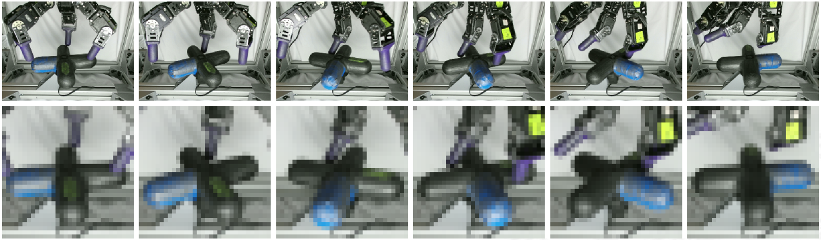
Advantages of using entropy

Some advantages of the maximum entropy objective:

- Policy explores more widely
- Learn multiple modes of near-optimal behavior, more robust
- Significantly speeds up learning

Progress

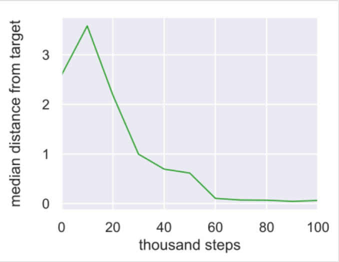
Dexterous hand manipulation

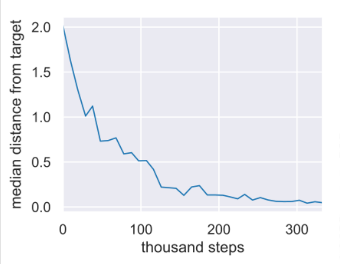


[9]

- 3-finger hand, 9 degrees of freedom
- Goal: Rotate valve into target position
- Learns directly from RGB images via CNN features
- Challenging due too complex hand and end-to-end perception
- 20 hours of real-world training

Dexterous hand manipulation



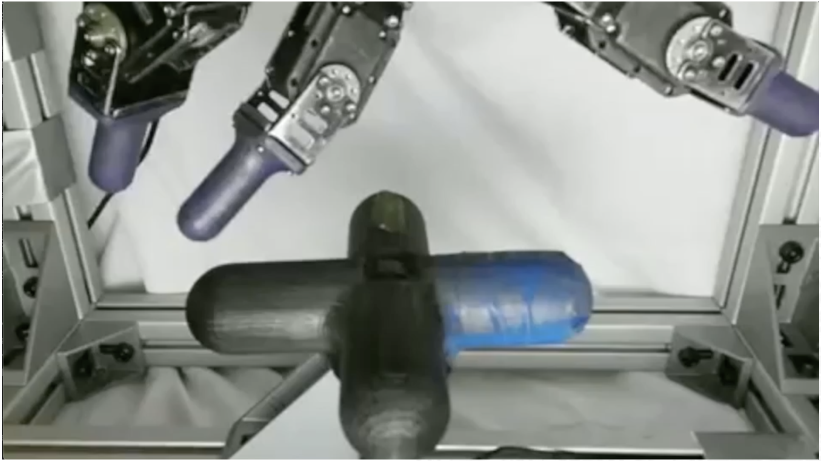


[9]

Alternative mode:

- Use valve position directly
- 3 hours of real-world training
- Substantially faster than competition on same tasks (PPO, 7.4 hours [10])

Dexterous hand manipulation



[11]

Simulated Benchmark

Comparison of SAC against other state of the art algorithms:

- DDPG, 2015: Off-policy, model-free, sample-efficient [12]
- TD3, 2018: Extension of DDPG [13]
- PPO, 2017: On-policy (relatively efficient), model-free [14]

Simpler and complex environments:

- Hopper-v2 (2D), Walker2D-v2 (2D), HalfCheetah-v2 (2D), Ant-v2 (3D)

- Humanoid-v2 (3D), Humanoid (rllab, 3D)

Simulated Benchmark

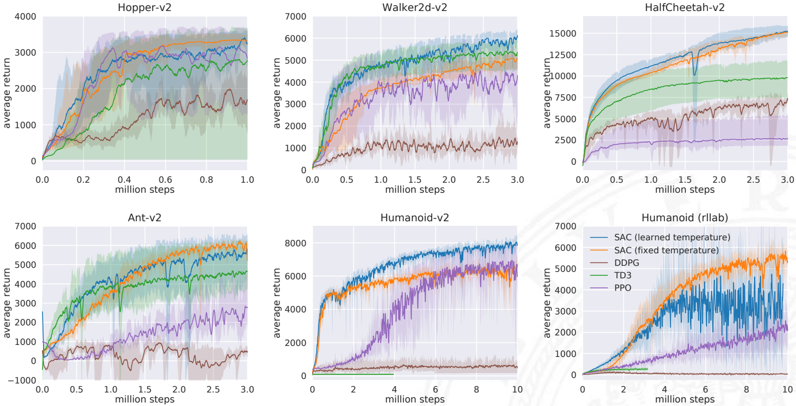


Figure taken from [9]

- Comparable to baseline on simple tasks
- Exceeds baseline on challenging tasks

Progress

Wrap-up & Conclusion

Soft actor-critic in a nutshell:

- Off-policy (higher sample efficiency)
- Model-free (almost necessity for real-world robotics)
- Training in simulation preferable, but still problematic
- Exploits entropy framework

Take-away:

- Can learn directly in real-world
- Can learn from raw sensory input (end-to-end)
- Entropy significantly speeds up learning
- Comparable to state of the art on simple tasks
- Exceeds state of the art on complex tasks

Question time

Thanks for your attention :)

References

[1] Xue Bin Peng et al. "DeepMimic". In: *ACM Transactions on Graphics* 37.4 (July 2018), pp. 1–14. ISSN: 0730-0301. DOI: 10.1145/3197517.3201311. URL: http://dx.doi.org/10.1145/3197517.3201311.

[2] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (Feb. 2015), pp. 529–533. ISSN: 00280836. URL: http://dx.doi.org/10.1038/nature14236.

[3] David Silver et al. "Mastering the game of go without human knowledge". In: *nature* 550.7676 (2017), pp. 354–359.

[4] Oriol Vinyals et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning". In: *Nature* 575.7782 (2019), pp. 350–354.

References (cont.)

[5] Shixiang Gu et al. “Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates”. In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 3389–3396.

[6] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Second. The MIT Press, 2018. URL: http://incompleteideas.net/book/the-book-2nd.html.

[7] S. Gu et al. “Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. May 2017, pp. 3389–3396. DOI: 10.1109/ICRA.2017.7989385.

[8] Tuomas Haarnoja et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". In: *arXiv preprint arXiv:1801.01290* (2018).

[9] Tuomas Haarnoja et al. "Soft actor-critic algorithms and applications". In: *arXiv preprint arXiv:1812.05905* (2018).

[10] Henry Zhu et al. "Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 3651–3657.

[11] *Soft Actor-Critic Project Website*. `https://sites.google.com/view/sac-and-applications`. Accessed: 2020-01-05.

References (cont.)

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

[13] Scott Fujimoto, Herke van Hoof, and David Meger. "Addressing function approximation error in actor-critic methods". In: *arXiv preprint arXiv:1802.09477* (2018).

[14] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

Value-based versus policy-based methods

So far, Value-based methods:

- Learn value-function (Q)
- Select actions based on learned value function
- Policies highly depend on value function

Alternatively, Policy-based methods:

- Learn parameterized policy
- No value function required, use total reward obtained from each action
- Can deal with continuous state and actions spaces
- However, requires complete transitions (Monte-Carlo)

Actor-critic methods

Why not use both?

- Learn policy (*actor*)
- Learn value-function (*critic*), approximating true value-function
- Basis for most recent RL algorithms

At each time-step (TD-approach):

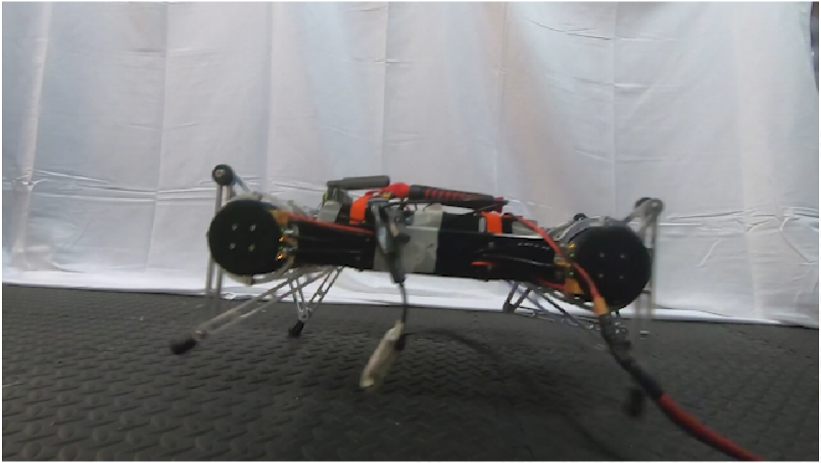
- Adjust critic to fit value-function
- Update actor to new critic
- This is the classical generalized policy iteration (GPI) algorithm
- Not possible for purely policy-based methods ()

Quadrupedal locomotion

Learning quadrupedal walking gaits:

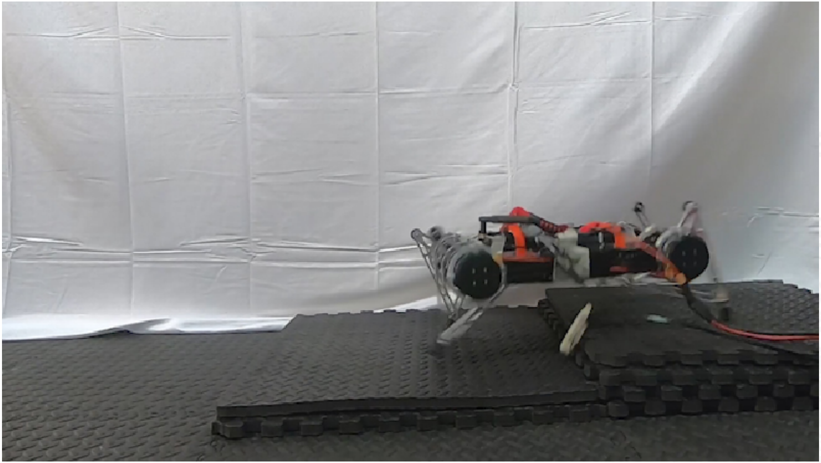
- Learning directly in real-world
- Some reward-engineering
- Walking learned within 2 hours of training
- First example of DRL on quadrupedal locomotion without any pretraining
- SAC policies are robust, generalizes well to unseen environment

Quadrupedal locomotion



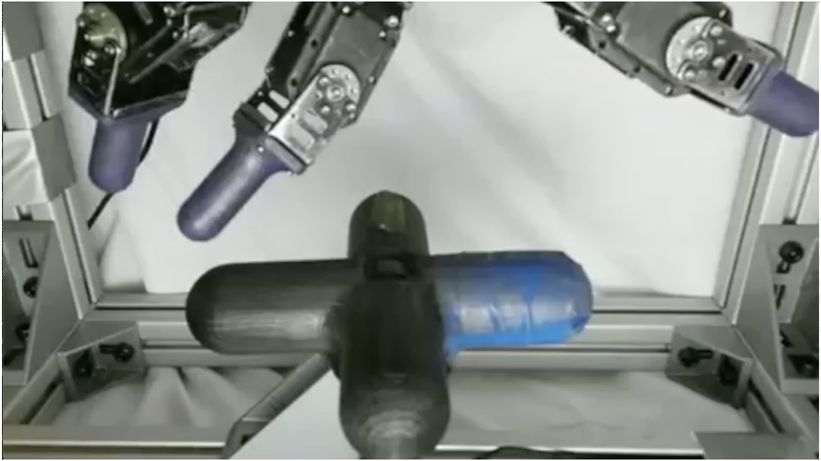
[11]

Quadrupedal locomotion



[11]

Dexterous hand manipulation



[11]