Reinforcement Learning of Motor Skills with Policy Gradients

Ruiming Zhang, Jieneng Chen, Karl Ylvisaker, Maxime Pasteur

{amber.zhang, Jieneng.chen, Karl.ylvisaker, Maxime.pasteur}@tum.de

Problem Setting

- Task: Learning complex motor skills with an anthropomorphic robot arm using reinforcement learning.
- Typical characteristics:
  - high-dimensional and continuous space and action space
  - has to be model-free
  - high degrees of freedom cannot deal with parameterised policies

Problem statement in Mathematics

- The general goal is to optimize the policy parameters $\theta \in R^K$ so that the expect return

$$J(\theta) = \frac{1}{a_\Sigma} E\{\sum_{k=0}^{H} a_k r_k\} \tag{1}$$

is maximized. a_k denote the time-step dependent weighting factors.

Policy Gradient Method

- The policy parameter θ is updated at each time step by:

$$\theta_{m+1} = \theta_m + \alpha_m \nabla_\theta J(\theta) \tag{2}$$

where $\alpha_m \in R^+$ denotes a learning rate.

- The main problem is obtaining a good estimator of the gradient $\nabla_\theta J|_{\theta=\theta_m}$.

Likelihood Ratio Method

- Use τ to represent a real generated trajectory, $\tau \sim p_\theta(\tau) = p(\tau|\theta)$, with rewards $r(\tau) = \sum_{k=0}^{H} a_k r_k$. Then the expect return of a policy can be written as an expectation over all possible trajectories:

$$J(\theta) = \int p_\theta(\tau) r(\tau) d\tau \tag{3}$$

- Subsequently, the gradient can be rewritten by:

$$\nabla_\theta J(\theta) = \int \nabla_\theta p_\theta(\tau) r(\tau) d\tau = \int p_\theta(\tau) \nabla_\theta log p_\theta(\tau) r(\tau) d(\tau) = E\{\nabla_\theta log p_\theta(\tau) r(\tau)\}$$

- The derivative $\nabla_\theta log p_\theta(\tau)$ can be computed by:

$$\nabla_\theta log p_\theta(\tau) = \sum_{k=0}^{H} \nabla_\theta log \pi_\theta(u_k|x_k) \tag{4}$$

- A constant baseline can be inserted since it has no effect to the derivative. Usually b is chosen with the goal to minimize the variance of the gradient estimator. and results in the final form of the gradient estimator:

$$\nabla_\theta J(\theta) = \left\langle \left(\sum_{k=0}^{H} \nabla_\theta log \pi_\theta(u_k|x_k) \right) \left(\sum_{l=0}^{H} a_l r_l - b \right) \right\rangle \tag{5}$$

Natural Actor-Critic Algorithm

- We first introduce second-order Taylor expansion to approximate the closeness of two distribution, i.e. the amount of change of the policy:

$$d_{KL}(p_\theta(\tau) \;||\; p_{\theta+\Delta_\theta}(\tau)) \approx \frac{1}{2} \Delta\theta^T F_\theta \Delta\theta \tag{6}$$

where

$$F_\theta = \int p_\theta(\tau) \nabla log p_\theta(\tau) \nabla log p_\theta(\tau)^T d\tau = \langle \nabla log p_\theta(\tau) \nabla log p_\theta(\tau)^T \rangle \tag{7}$$

is known as the Fisher information matrix.

Natural Gradient

- Assume that the amount of change is fixed using step size ε. Then the optimization problem can be described as:

$$\begin{aligned} &\max_{\Delta\theta} J(\theta + \Delta\theta) \approx J(\theta) + \Delta\theta^T \nabla_\theta J \\ &s.t.\ \varepsilon = d_{KL}(p_\theta(\tau) \;||\; p_{\theta+\Delta\theta}(\tau) \approx \tfrac{1}{2} \Delta\theta^T F_\theta \Delta\theta) \end{aligned} \tag{8}$$

and has the solution:

$$\Delta\theta = \alpha_n F_\theta^{-1} \nabla_\theta J \tag{9}$$

with $\alpha_n = [\varepsilon (\nabla J(\theta)^T F_\theta^{-1} \nabla J(\theta))^{-1}]^{\frac{1}{2}}$.

- The item $\widetilde{\nabla}_\theta J(\theta) = \Delta\theta / \alpha_n$ is called the natural gradient and we will use it to replace the original gradient $\nabla_\theta J$ which represents the steepest ascend in order to obtain a faster convergence and stable update.

Compatible Function Approximation

- Use a compatible function approximation parameterized by ω to repalce the critic term:

$$(\nabla_\theta log \pi(u|x))^T \omega = Q^\pi(x, u) - b^\pi(x) \tag{10}$$

- Thus we derive an estimte of the policy gradient as:

$$\nabla_\theta J(\theta) = \int_x d^\pi(x) \int_u \nabla_\theta \pi(u|x) \nabla_\theta log \pi(u|x)^T du dx \omega = \int_x d^\pi(x) \hat{G}_\theta(x) dx \omega = G_\theta \omega \tag{11}$$

- The left-undecided Fisher information matrix in last section can be determined through sampling:

$$\begin{aligned} F_\theta &= -\langle \nabla_\theta^2 log p(\tau_{0:H}) \rangle = -\left\langle \sum_{k=0}^{H} \nabla_\theta^2 log \pi(u_H|x_H) \right\rangle \\ &= -\int_x d_H^\pi(x) \int_u \pi(u|x) \nabla_\theta^2 log pi(u|x) du dx = G_\theta \end{aligned} \tag{12}$$

- Thus the natural gradient can be simply computed as $\widetilde{\nabla}_\theta J(\theta) = F_\theta^{-1} G_\theta \omega = \omega$. Therefore the resulting policy improvement step becomes $\theta_{i+1} = \theta_i + \alpha\omega$.

Episodic Natural Actor-Critic Algorithm

- The Bellman equation can be witten in terms of the advantage function and the state-value function:

$$Q^\pi(x, u) = A^\pi(x, u) + V^\pi(x) = r(x, u) + \gamma \int_x p(x'|x, u) V^\pi(x') dx' \tag{13}$$

- Inserting $A^\pi(x, u)$ as the compatible value approximation term and $V^\pi(x)$ an appropriate basis function representation $\phi(x)^T v$ we have:

$$\nabla_\theta log \pi(u_t|x_t)^T \omega + \phi(x_t)^T v = r(x_t, u_t) + \gamma \phi(x_{t+1}^T) v + \varepsilon(x_t, u_t, x_{t+1}) \tag{14}$$

- For episodic tasks we can derive a simplified form:

$$\sum_{t=0}^{H} a_t \nabla log \pi(u_t, x_t)^T w + J_0 = \sum_{t=0}^{H} a_t r(x_t, u_t) \tag{15}$$

- This means for non-stochastic tasks we can obtain a natural gradient after $\dim\theta + 1$ roll-outs using least-squares regression:

$$\begin{bmatrix} \omega \\ J_0 \end{bmatrix} = (\Psi^T \Psi)^{-1} \Psi^T R \tag{16}$$

with

$$\Psi_i = \left[\sum_{t=0}^{H} a_t \nabla log \pi(u_t, x_t)^T, 1 \right] \tag{17}$$

$$R_i = \sum_{t=0}^{H} a_t r(x_t, u_t) \tag{18}$$

- In order to take time-variance rewards significantly better into account, we use a time-variant average rewards $\overline{r} = [\overline{r}_1, \overline{r}_2, ..., \overline{r}_K$ and then we have to solve:

$$\begin{bmatrix} g_{eNACn} \\ \overline{r} \end{bmatrix} = \begin{bmatrix} F_2 & \overline{\Phi} \\ \overline{\Phi}^T & mI_H \end{bmatrix} \begin{bmatrix} g \\ \overline{r} \end{bmatrix} \tag{19}$$

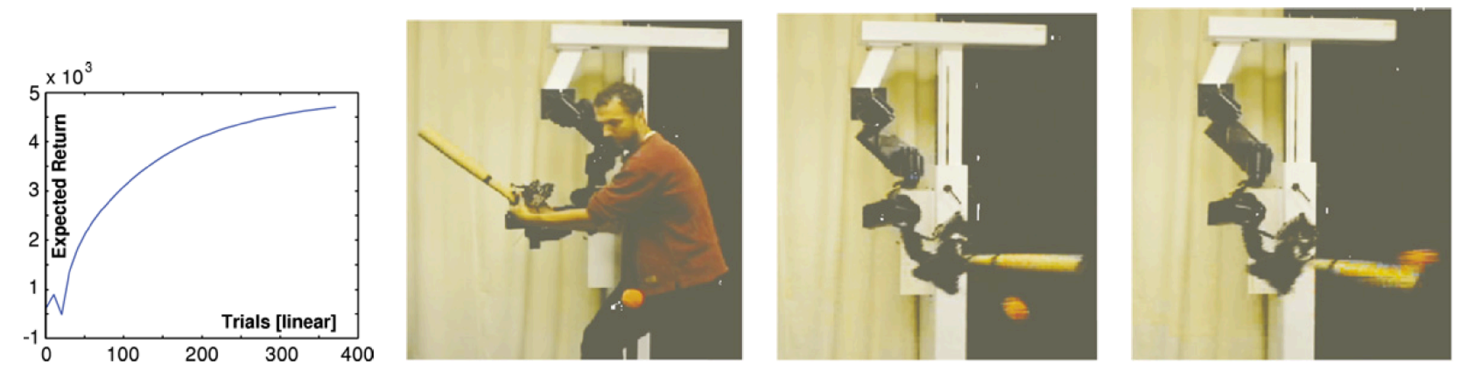
and finally obtained:

$$b = Q^{-1}(\overline{r} - \overline{\Phi}^T F_2^{-1} g) \tag{20}$$

with $Q^{-1} = m^{-1}(I_n + \overline{\Phi}^T (mF_2 - \overline{\Phi}\overline{\Phi}^T)^{-1} \overline{\Phi})$ and Φ is the eligibility matrix.

Experiments & Conclusion

- Task: Applying the Episodic Natural Actor-Critic to a Sarcos Master Arm to hit a baseball.
- Figure(a) shows the average of the reward. The robot fail to reproduce the behavior at first. Subsequently the accuracy of the hitting angle is improved to hit properly after 200-300 trials.



(a) Performance. (b) Imitation learning. (c) Initial reproduction. (d) After reinforcement learning.

- Conclusion: The example of motor primitive learning for baseball underlines the efficiency of natural gradient methods for complex movement systems.

Research Group for Geometric Optimization and Machine Learning
Technische Universität München, Germany