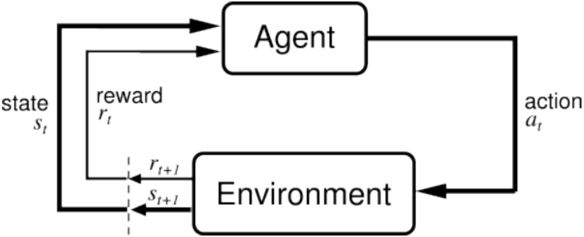
Actor-Critic Algorithm with Approximate Natural Policy Gradient

Evgenii Nikishin, Iurii Kemaev, Maksim Kuznetsov

22.12.2017

Reinforcement Learning (RL) setting

Most of RL methods consider Markov Decision Process (MDP):



- Agent actions $A_t \in \mathcal{A}$
- Environment states $S_t \in \mathcal{S}$
- Reward $R_t \in \mathbb{R}$
- Agent policy $A_t \sim \pi(a|s)$
- State transitions $S_{t+1}, R_{t+1} \sim p(s', r|s, a)$

Optimal policy

Total return is defined as the sum of discounted rewards:

$$G_0 = R_1 + \gamma R_2 + \gamma^2 R_3 + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{k+1}, \quad 0 \leq \gamma \leq 1$$

Optimal policy maximizes the expected discounted return:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi}\left[G_0\right] = \arg\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k R_{k+1}\right]$$

Value functions:

- $v_\pi(s) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right]$
- $q_\pi(s, a) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a\right]$

Policy Gradient theorem

Parametrization of policy with differentiable family of functions (e.g. linear functions, neural networks):

$$\pi(a|s) = \pi_\theta(a|s)$$

Objective:

$$J(\theta) = \mathbb{E}_{\pi_\theta}[G_0]$$

...and its gradient:

$$\begin{aligned}\nabla_\theta J(\theta) = & \sum_s \mu(s) \sum_a q_{\pi_\theta}(s, a) \nabla_\theta \log \pi_\theta(s, a) = \\ & = \mathbb{E}_{\pi_\theta}\left[q_{\pi_\theta}(S_t, A_t) \nabla_\theta \log \pi_\theta(A_t|S_t)\right]\end{aligned}$$

Advantage Actor-Critic (A2C) algorithm

Challenges:

- $q_{\pi_\theta}(s, a)$ is unknown
- both $q_{\pi_\theta}(S_t, A_t)$ and $\log \pi_\theta(A_t|S_t)$ are unbounded
  $\Rightarrow$ high variance of gradient estimates

Note:

- $v_{\pi_\theta}(s) = \sum_a \pi_\theta(a|s) q_{\pi_\theta}(s, a) = \mathbb{E}_a \left[q_{\pi_\theta}(s, a)\right]$
- $q_{\pi_\theta}(S_t, A_t) = r_{t+1} + \gamma \mathbb{E}_{\pi_\theta} \left[v_{\pi_\theta}(S_{t+1})\right]$

Advantage Actor-Critic (A2C) algorithm

Approaches:

- Subtract average q function and rewrite in terms of advantage function:

$$\begin{aligned}\nabla_\theta J(\theta) &= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(A_t|S_t)\left(q_{\pi_\theta}(S_t, A_t) - v_{\pi_\theta}(S_t)\right)\right] = \\ &= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(A_t|S_t)\left(R_t + \gamma v_{\pi_\theta}(S_{t+1}) - v_{\pi_\theta}(S_t)\right)\right]\end{aligned}$$

- v function is still unknown $\Rightarrow$ parametrize it (critic):

$$v_{\pi_\theta}(s) \approx v_\omega(s)$$

and optimize w.r.t. objective

$$\|v_\omega(S_t) - G_t\|^2$$

Natural gradient

Update rule:

$$\theta_{t+1} = \theta_t + G^{-1}(\theta_t)\nabla F(\theta_t),$$

$F(\theta_t)$ — objective, $G(\theta_t)$ — Fisher matrix:

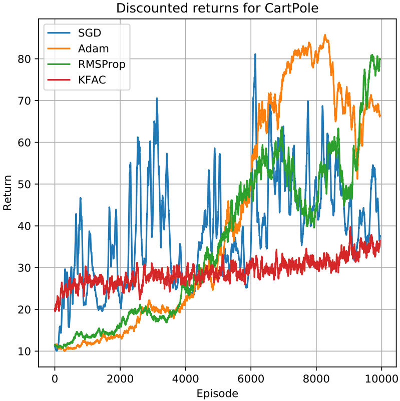
$$G(\theta) = \mathbb{E}[\nabla_\theta \log \pi_\theta(A_t|S_t) \cdot (\nabla_\theta \log \pi_\theta(A_t|S_t))^\top] = \mathbb{E}[\nabla_\theta L \cdot \nabla_\theta L^\top]$$

Consider vector of activations a. For a current layer denote $s = Wa$.
$\nabla_W L = (\nabla_s L)a^\top$

Kronecker Factored Approximate Curvature (K-FAC)

Consider vector of activations a. For a current layer denote $s = Wa$.
$\nabla_W L = (\nabla_s L)a^\top$

$$
\begin{aligned}
G(W) &= \mathbb{E}\left[vec\{\nabla_W L\}vec\{\nabla_W L\}^\top\right] = \\
&= \mathbb{E}\left[aa^\top \otimes \nabla_s L(\nabla_s L)^\top\right] \approx \mathbb{E}\left[aa^\top\right] \otimes \mathbb{E}\left[\nabla_s L(\nabla_s L)^\top\right]
\end{aligned}
$$



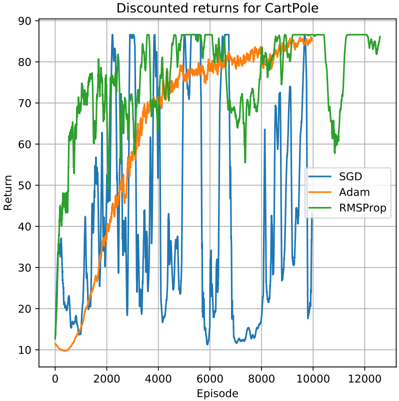


Figure: Left: A2C. Right: A3C

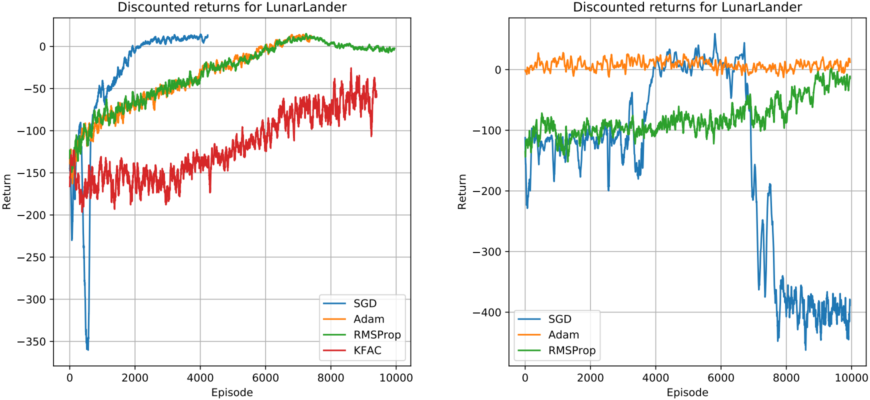


Figure: Left: A2C. Right: A3C

Experiments

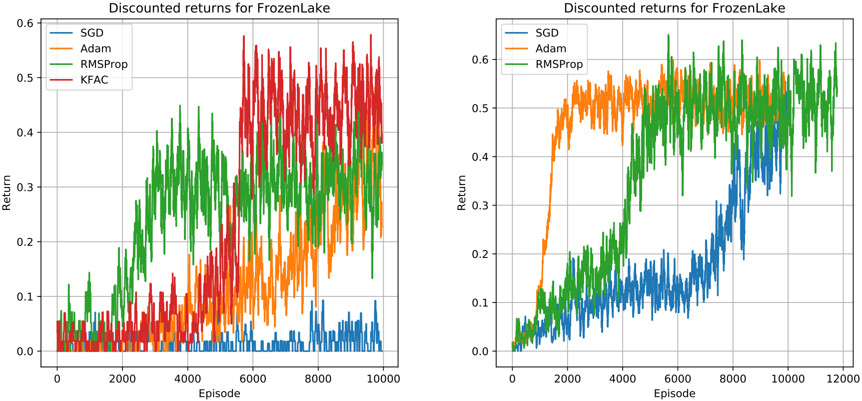


Figure: Left: A2C. Right: A3C

What was done

- Evgenii Nikishin: idea, simple policy gradient algorithm, presentation
- Iurii Kermaev: Actor-Critic Algorithm, its parallel version, experiments
- Maksim Kuznetsov: K-FAC optimizer, experiments, debug

Original paper:
Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation