Proximal Policy Optimization (PPO)

A. Opris, A. Santamaria, M. Kreutz

T3 Drive

Schwächen von Policy Gradient Methoden (PGM)

- Schwer gute Resultate mit PGM zu erreichen
- PGM reagieren empfindlich bei der Wahl der Stepsize
- Ist die Stepsize zu
  - klein gewählt, ist der Fortschritt hoffnungslos langsam
  - groß gewählt und der Input verrauscht, dann führt das starken Einbrüchen in der Performance
- Ineffizientes Sampling, da hier Millionen oder auch Milliarden Timesteps benötigt werden um einfache Aufgaben zu erlernen

Vanilla Policy Gradient (VPG) | Trust Region Policy Optimization (TRPO) | Proximal Policy Optimization (PPO)

$$\nabla_\theta J(\pi_\theta) = \mathop{\mathrm{E}}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) A^{\pi_\theta}(s_t, a_t)\right]$$

Vanilla Policy Gradient (VPG)

Trust Region Policy Optimization (TRPO)

Proximal Policy Optimization (PPO)

$$\mathcal{L}(\theta_k, \theta) = \underset{s,a\sim\pi_{\theta_k}}{\mathrm{E}} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a) \right]$$

Nebenbedingung:

$$\bar{D}_{KL}(\theta||\theta_k) = \underset{s\sim\pi_{\theta_k}}{\mathrm{E}} \left[D_{KL}\left(\pi_\theta(\cdot|s)||\pi_{\theta_k}(\cdot|s)\right)\right] \leq \delta$$

Vanilla Policy Gradient (VPG) | **Trust Region Policy Optimization (TRPO)** | Proximal Policy Optimization (PPO)

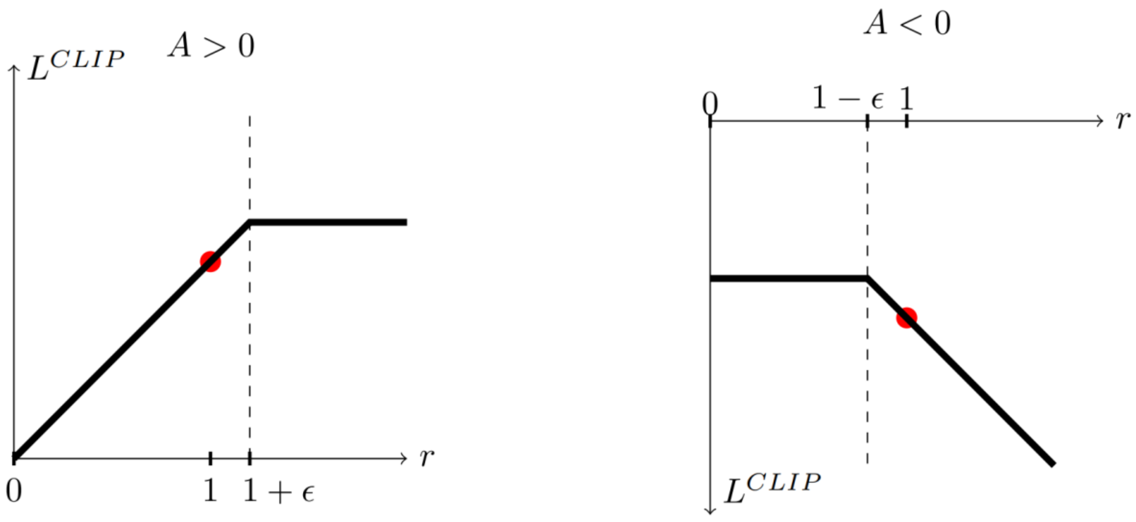
$$\mathcal{L}(\theta_k, \theta) = \underset{s,a\sim\pi_{\theta_k}}{\mathrm{E}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a)\right]$$

Nebenbedingung:

$$\bar{D}_{KL}(\theta||\theta_k) = \underset{s\sim\pi_{\theta_k}}{\mathrm{E}}\left[D_{KL}\left(\pi_\theta(\cdot|s)||\pi_{\theta_k}(\cdot|s)\right)\right] \leq \delta$$

Vanilla Policy Gradient (VPG) → Trust Region Policy Optimization (TRPO) → **Proximal Policy Optimization (PPO)**

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$



Probability Ratio:

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$$

Estimated Advantage Funktion:

$$\hat{A}_t = -V(s_t) + r_t + \gamma r_{t+1} + \cdots + \gamma^{T-t+1} r_{T-1} + \gamma^{T-t} V(s_T)$$

Vanilla Policy Gradient (VPG)

Trust Region Policy Optimization (TRPO)

Proximal Policy Optimization (PPO)

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t\left[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)\right]$$

Entropy Bonus für hinreichende Exploration:

S

Squared-error loss:

$L_t^{VF} = (V_\theta(s_t) - V_t^{\text{targ}})^2$

Vanilla Policy Gradient (VPG)

Trust Region Policy Optimization (TRPO)

Proximal Policy Optimization (PPO)

$$L^{KLPEN}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t - \beta\,\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]\right]$$

$$d = \hat{\mathbb{E}}_t[\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]]$$

$$d < d_{\text{targ}}/1.5,\ \beta \leftarrow \beta/2$$

$$d > d_{\text{targ}} \times 1.5,\ \beta \leftarrow \beta \times 2$$

Vanilla Policy Gradient (VPG)

Trust Region Policy Optimization (TRPO)

Proximal Policy Optimization (PPO)

No clipping or penalty: $L_t(\theta) = r_t(\theta)\hat{A}_t$

Clipping: $L_t(\theta) = \min(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta)), 1-\epsilon, 1+\epsilon)\hat{A}_t$

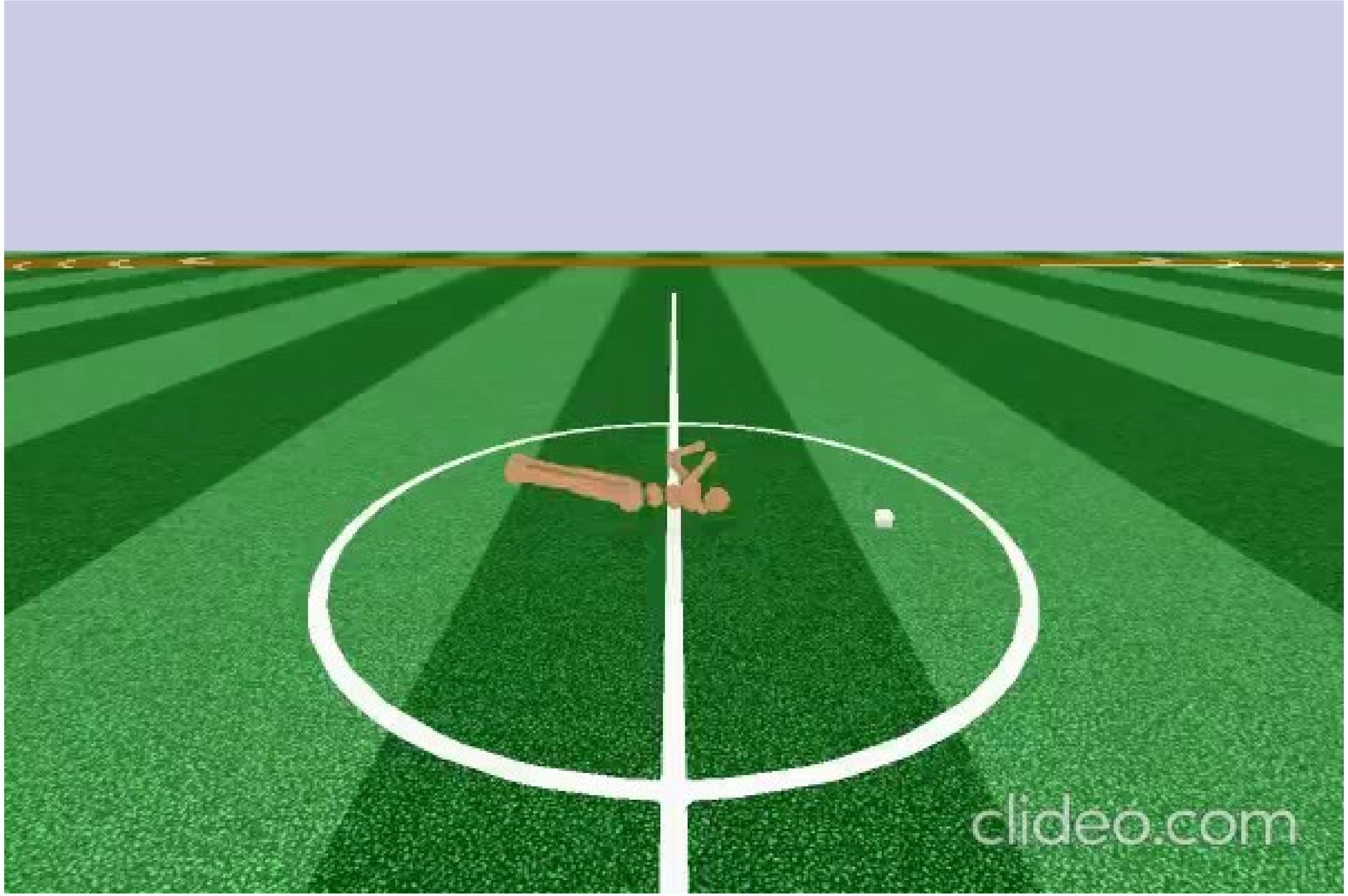
KL penalty (fixed or adaptive) $L_t(\theta) = r_t(\theta)\hat{A}_t - \beta\,\mathrm{KL}[\pi_{\theta_{\mathrm{old}}}, \pi_\theta]$

algorithm	avg. normalized score
No clipping or penalty	-0.39
Clipping, $\epsilon = 0.1$	0.76
Clipping, $\epsilon = 0.2$	**0.82**
Clipping, $\epsilon = 0.3$	0.70
Adaptive KL $d_{\mathrm{targ}} = 0.003$	0.68
Adaptive KL $d_{\mathrm{targ}} = 0.01$	0.74
Adaptive KL $d_{\mathrm{targ}} = 0.03$	0.71
Fixed KL, $\beta = 0.3$	0.62
Fixed KL, $\beta = 1.$	0.71
Fixed KL, $\beta = 3.$	0.72
Fixed KL, $\beta = 10.$	0.69

Tabelle: Basierend auf sieben simulierte Roboteraufgaben in OpenAI Gym mit MuJoCo Physics Engine

Proximal Policy Optimization (PPO) in Action

Roboschool (trained by OpenAI)



Quellen

- https://openai.com/blog/openai-baselines-ppo/
- https://spinningup.openai.com/en/latest/algorithms/vpg.html
- https://spinningup.openai.com/en/latest/algorithms/ppo.html
- https://spinningup.openai.com/en/latest/algorithms/trpo.html
- https://jonathan-hui.medium.com/rl-proximal-policy-optimization-ppo-explained-77f014ec3f12
- https://arxiv.org/pdf/1707.06347.pdf
- https://arxiv.org/pdf/1502.05477.pdf
- https://towardsdatascience.com/understanding-and-implementing-proximal-policy-optimization-schulman-et-al-2017-9523078521ce
- http://rail.eecs.berkeley.edu/deeprlcourse-fa17/f17docs/lecture_13_advanced_pg.pdf
- https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence