PPO in OpenAI Gym

David Weiler

Seminar Deep Learning

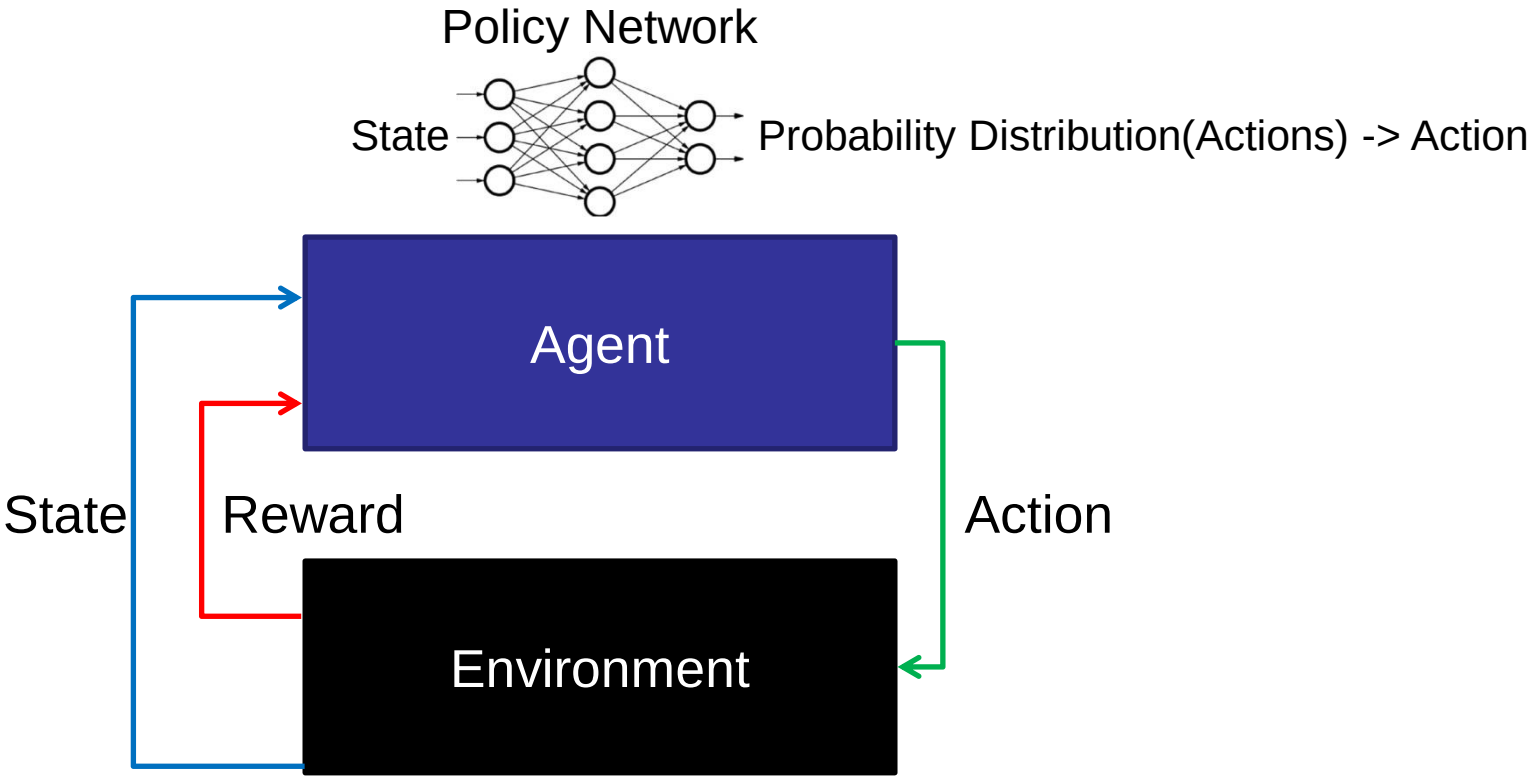
Hochschule Offenburg

Elektrotechnik und Informationstechnik

Inhalt

- Reinforcement Learning mit Policy Gradient
  - Überblick
  - Probleme
- Proximal Policy Optimization
- OpenAi Gym
- Resultate

Reinforcement Learning mit Policy Gradient



- Policy Gradient: Interagiere eine Weile mit der Umgebung
  - erhöhe Wahrscheinlichkeit der Actions die zu positivem Reward führten
  - veringere Wahrscheinlichkeit der Actions die zu negativem Reward führten

Policy Gradient: Problems

- Empfindlichkeit gegenüber learning rate (Hyperparameter)
  - Zu klein -> langsamer Fortschritt
  - Zu groß -> schlechte policy -> nächste Traingsdaten werden unter schlechten Bedingungen gesammelt

  -> nur eine Updates pro Trainingsdaten -> sample inefficent
- Bisherige Lösungen
  - Zusätzliche komplexere/teurere Berechnungen

    -> PPO: limitiere Policy Updates

Proximal Policy Optimization (PPO)

War die Action besser oder schlechter als erwartet?

Algorithm 1 PPO

for iteration=$1, 2, \ldots$ **do**
  for actor=$1, 2, \ldots, N$ **do**
    Run policy $\pi_{\theta_{\text{old}}}$ in environment for T timesteps
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
  end for
  Optimize surrogate L wrt θ, with K epochs and minibatch size $M \leq NT$
  $\theta_{\text{old}} \leftarrow \theta$
end for

Optimierungsschritt

Proximal Policy Optimization (PPO)

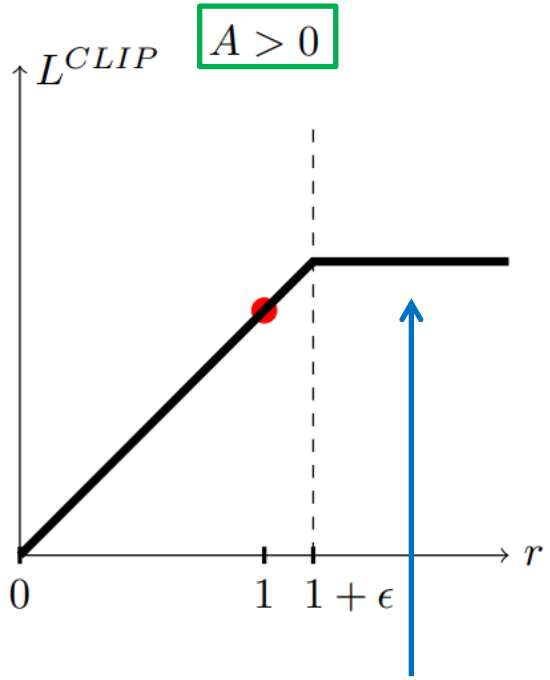
$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$$

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

πθ(a|s) : Die Wahrscheinlichkeit die Aktion a im State s auszuwählen

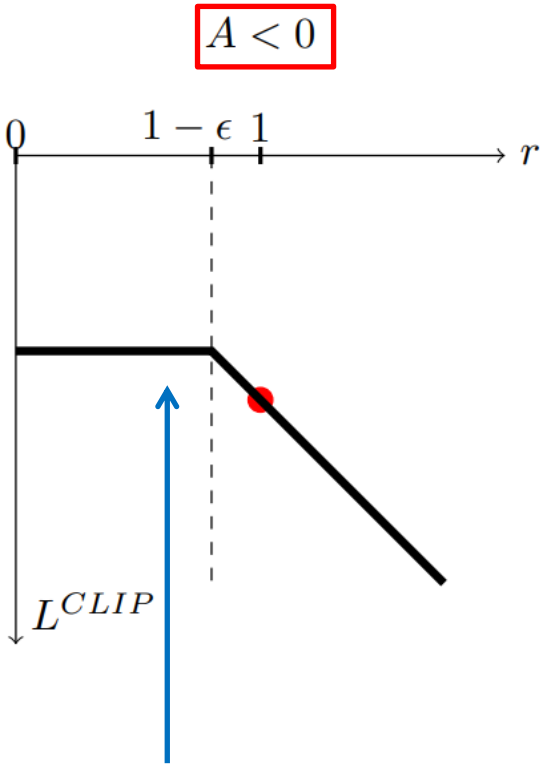
Proximal Policy Optimization (PPO)

Positiver Advantage:
Action war besser als erwartet…



r positiv:
… und wurde wahrscheinlicher

Negativer Advantage:
Action war schlechter als erwartet…



r negativ:
… und wurde weniger wahrscheinlich

Proximal Policy Optimization (PPO)

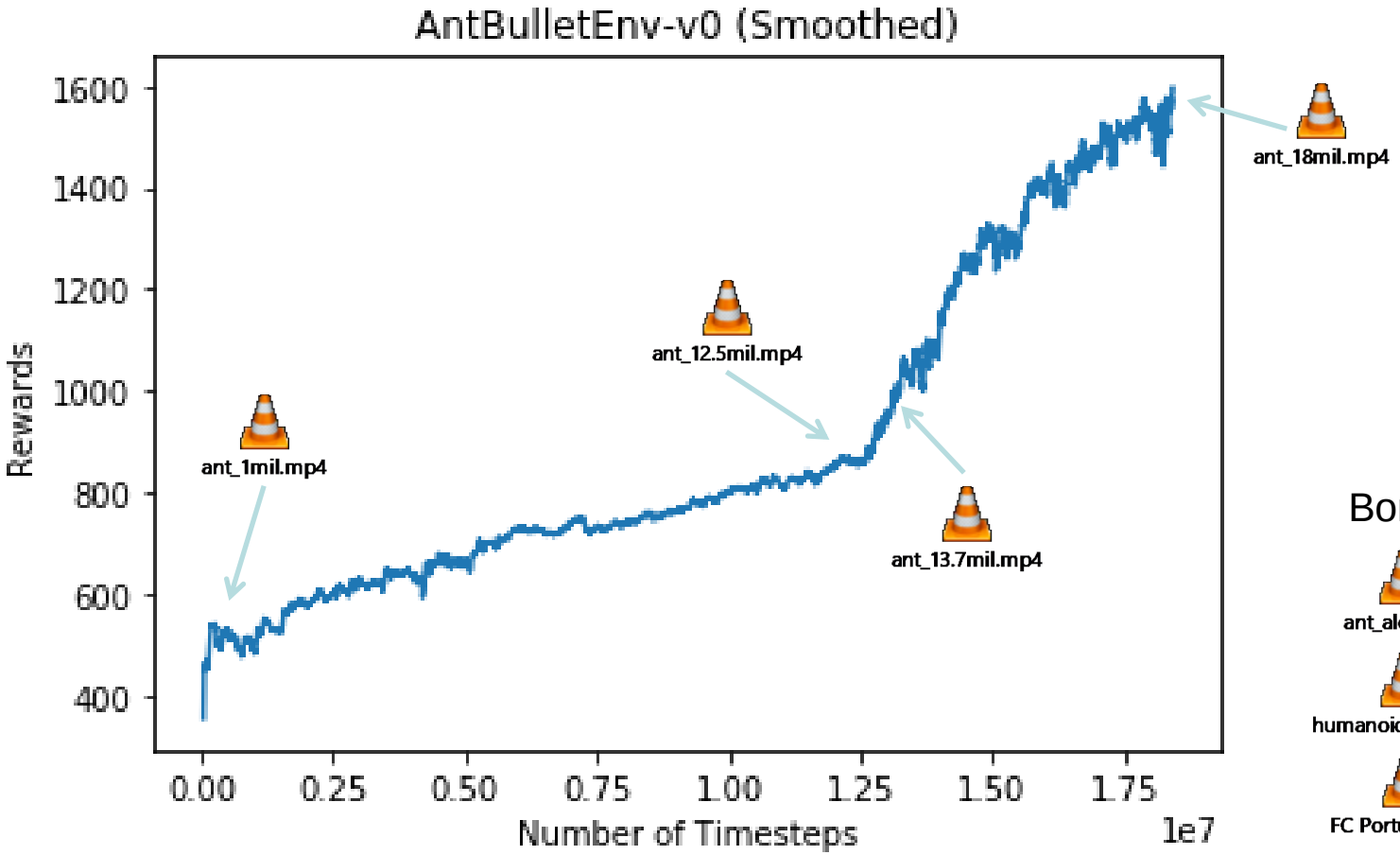
Algorithm 1 PPO

for iteration=$1, 2, \ldots$ **do**
  for actor=$1, 2, \ldots, N$ **do**
    Run policy $\pi_{\theta_{\text{old}}}$ in environment for T timesteps
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
  end for
  Optimize surrogate L wrt θ, with K epochs and minibatch size $M \leq NT$
  $\theta_{\text{old}} \leftarrow \theta$
end for

OpenAi Gym

- Toolkit zum Entwickeln und Vergleichen von reinforcement learning Algorithmen
- Mehrere Environments (2D, 3D, Atari Spiele) mit einheitlichem Interface
- Nutzt TensorFlow

Results (3D-Environment „Ant“)



Ressources

- GitHub (Jupyter notebook + videos): https://github.com/dav-92/PPO

- PPO Paper: https://arxiv.org/pdf/1707.06347.pdf
- OpenAI Gym: https://gym.openai.com/
- Reinforcement learning Algorithms: https://github.com/hill-a/stable-baselines

Basic Policy Gradient

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta(a_t \mid s_t) \hat{A}_t \right]$$