Seminar in Deep Reinforcement Learning

Introduction
Reinforcement Learning...
Why you should NOT use reinforcement learning...

WHENEVER SOMEONE ASKS ME IF RL WORKS, I TELL THEM IT DOESN'T

AND 70% OF THE TIME, I'M RIGHT
Why you should NOT use reinforcement learning...

Reward Engineering can be hard - small auxiliary rewards might become the main focus of the agent.
Why is reinforcement learning promising?

- Humans are reinforcement learners
- Agents can outperform teachers
- Can model any (easy verifiable) task
Disclaimer: This is a seminar...

(almost) no basics

participation required
Format

- Assigned topics
- 35 min presentation
- 10 min facilitated discussion
- Voluntary coding challenge

Grade =
presentation + active participation (+ challenge)
Reinforcement Learning...
On Policy vs. Off-Policy vs. Batch-Policy Learning

Agent

Learn Off-Policy

State Reward

Action

Environment
Deep reinforcement learning in continuous action spaces
Hierarchical deep reinforcement learning

Action
“move to x”
vs
“move forward”

Agent

State
Reward

Environment
- Deep reinforcement learning and stochastic planning in games
- Model based vs. model free deep reinforcement learning
- Deep reinforcement learning in partial observability
Multi-Armed Bandits

Agent

Context

Action

Reward

Environment
Non-differentiable optimization

Agent

State Reward

Action

Environment

non-differentiable
Meta-Learning

Agent

State
Reward

Action

non-differentiable

Earth
What do I have to remember?
Task?

Maximize the discounted cumulative reward in each episode by finding a good policy
Task?

Maximize the discounted cumulative reward in each episode by finding a good policy
Task?

Maximize the **discounted cumulative reward** in each episode by finding a good policy

\[ R = \sum_t \gamma^t r_t \]

\[ \gamma \in (0, 1] \]
Maximize the discounted cumulative reward in each episode by finding a good policy

\[ R = \sum_{t=0}^{T_e} \gamma^t r_t \]

\[ \gamma \in (0, 1] \]
Task?

Maximize the discounted cumulative reward in each episode by finding a good policy

\[ R^\pi = \sum_{t=0}^{T_e} \gamma^t r_t \]

\[ \pi(a|s_t) = Pr(a|s_t) \]
How?

Estimate remainder of $R^\pi$ in each state $s_t$

$$V^\pi(s_t) = \mathbb{E}_\pi \left[ \sum_{t'=t}^{T^e} \gamma^{t'-t} r_{t'} \right]$$

$$Q^\pi(s_t, a_t) = r_t|a_t + \mathbb{E}_{\pi|a_t} \left[ \sum_{t'=t+1}^{T^e} \gamma^{t'-t} r_{t'} \right]$$
\[ V^\pi(s_t) = \mathbb{E}_\pi \left[ \sum_{t'=t}^{T_e} \gamma^{t'-t} r_{t'} \right] \]
\[ = \mathbb{E}_\pi [r_t] + \gamma \mathbb{E}_\pi \left[ \sum_{t'=t+1}^{T_e} \gamma^{t'-t-1} r_{t'} \right] \]
\[ = \mathbb{E}_\pi [r_t] + \gamma V^\pi(s_{t+1}) \]
\[ Q^\pi(s_t, a_t) = r_t \mid a_t + \mathbb{E}_{\pi \mid a_t} \left[ \sum_{t'=t+1}^{T_e} \gamma^{t'-t} r_{t'} \right] \]

\[ = r_t \mid a_t + \gamma V^\pi(s_{t+1}) \]

\[ V^\pi(s_t) = \mathbb{E}_{a \sim \pi} \left[ Q^\pi(s_t, a) \right] \]
$\pi^{\text{greedy}}(a|s_t) = 1_{a=\max_{a'} Q^*(s_t,a')}$
Q-Learning

\[ Q^{\text{greedy}}(s_t, a_t) = r_t|a_t + \gamma \max_{a'} Q^{\text{greedy}}(s_{t+1}, a') \]

iff \( Q^{\text{greedy}} \equiv Q^* \)

\[ y(s_t, a_t) := r_t|a_t + \gamma \max_{a'} \tilde{Q}(s_{t+1}, a') \]

\[ \delta_{TD} = y(s_t, a_t) - \tilde{Q}(s_t, a_t) \]

\[ \rightarrow \text{minimize } \delta_{TD}^2 \]
Classical RL vs Deep RL

estimate $\tilde{Q}(s, a)$

$\forall (s, a) \in \mathcal{S} \times \mathcal{A}$

$\rightarrow$ approximate $\tilde{Q}(\cdot, \cdot)$
Human-level control through DRL (Mnih et al., 2015)
Deep Learning works for... RL?

- ...large data sets... many interactions
- ...with labeled data points... self-labeled
- ...which are iid

→ target network
→ replay buffer
Target Network

\[ y(s_t, a_t) := r_t | a_t + \gamma \max_{a'} \tilde{Q}_\theta^-(s_{t+1}, a') \]

\[ \delta_{TD} = y(s_t, a_t) - \tilde{Q}_\theta(s_t, a_t) \]

\[ \rightarrow \text{minimize } \delta_{TD}^2 \]
Replay Buffer

Learn from samples
Task?

Maximize the discounted cumulative reward in each episode by finding a good policy

\[ R^{\pi} = \mathbb{E}_{\pi} \left[ \sum_{t=0}^{T_e} \gamma^t r_t \right] \]

\[ \pi_{\theta}(a|s_t) = Pr(a|s_t; \theta) \]
\[ \max_{\theta} R^{\pi_{\theta}} \]

\[ \rightarrow \theta_{k+1} = \theta_k + \alpha \nabla_{\theta} R^{\pi_{\theta}} \]

\[ \nabla_{\theta} \mathbb{E}_{\tau \sim \pi_{\theta}} [R(\tau)] = ? \]
Policy Gradient Derivation

\[ \nabla \mathbb{E}_\pi [R(\tau)] = \nabla \int R(\tau) \pi(\tau) \, d\tau \]

\[ = \int R(\tau) \nabla \pi(\tau) \, d\tau \]

\[ = \int R(\tau) \pi(\tau) \frac{\nabla \pi(\tau)}{\pi(\tau)} \, d\tau \]

\[ = \int R(\tau) \pi(\tau) \nabla \log \pi(\tau) \, d\tau \]

\[ = \mathbb{E}_\pi [R(\tau) \nabla \log \pi(\tau)] \]
\[ \pi_{\theta}(\tau) = \mathcal{P}(s_0) \prod_{t=0}^{T_e} \pi_{\theta}(a_t \mid s_t)p(s_{t+1} \mid s_t, a_t) \]

\[ \rightarrow \nabla_\theta \log \pi_{\theta}(\tau) = \sum_{t=0}^{T_e} \nabla_\theta \log \pi_{\theta}(a_t \mid s_t) \]

\[ \mathbb{E}_\pi \left[ \mathcal{R}(\tau) \nabla \log \pi(\tau) \right] \]

\[ = \mathbb{E}_\pi \left[ (\sum_{t=0}^{T_e} \gamma^t r_t)(\sum_{t=0}^{T_e} \nabla \log \pi(a_t \mid s_t)) \right] \]
\[ \mathbb{E}_\pi \left[ \left( \sum_{t=0}^{T_e} \gamma^t r_t \right) \left( \sum_{t=0}^{T_e} \nabla \log \pi(a_t | s_t) \right) \right] \]

\[ = \mathbb{E}_\pi \left[ \sum_{t=0}^{T_e} \left( \sum_{t'=t}^{T_e} \gamma^{t'-t} r_{t'} \right) \nabla \log \pi(a_t | s_t) \right] \]

\[ = \mathbb{E}_\pi \left[ \sum_{t=0}^{T_e} V^\pi(s_t) \nabla \log \pi(a_t | s_t) \right] \]

\[ = \mathbb{E}_\pi \left[ \sum_{t=0}^{T_e} \left( V^\pi(s_t) - b \right) \nabla \log \pi(a_t | s_t) \right] \]

Causality

b independent of action
Asynchronous Methods for DRL (Mnih et al., 2016)
Asynchronous Methods for DRL (Mnih et al., 2016)

$$\max_{\theta} \mathbb{E}_{\pi_{\theta}} \left[ \sum_{t=0}^{T_e} (V^{\pi_{\theta}}(s_t) - b) \log \pi_{\theta}(a_t|s_t) \right]$$

$$b = \tilde{V}_\phi(s_t)$$

$$V^{\pi_{\theta}}(s_t) \approx \sum_{t'=t}^{t+n} \gamma^{t'-t} r_{t'} + \gamma^n \tilde{V}_\phi(s_{t+n})$$

A3C
Deep Learning works for... RL?

- ...large data sets...
  - many interactions
- ...with labeled data points...
  - self-labeled
- ...which are iid

  ➔ multi-step target
  ➔ self-labeled
  ➔ multiple actors
  ➔ entropy regularization
Multiple Actors

Learn from all rollouts
Entropy Regularization

... act as random as possible

$$\max_{\theta} \mathbb{E}_{\pi_{\theta}} \left[ \sum_{t=0}^{T_e} (V^{\pi_{\theta}}(s_t) - b) \log \pi_{\theta}(a_t | s_t) - \lambda \pi_{\theta}(a_t | s_t) \log \pi_{\theta}(a_t | s_t) \right]$$
DQN vs A3C

- Sample efficient vs sample inefficient
- Slow to train vs fast to train
- (Almost) deterministic vs stochastic
- Only 1 network vs 2 (1.5) networks
Coding Challenge

https://github.com/OliverRichter/DRL_Seminar_BlackJack
References


Asynchronous Methods for Deep Reinforcement Learning, Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, Koray Kavukcuoglu, ICML 2016