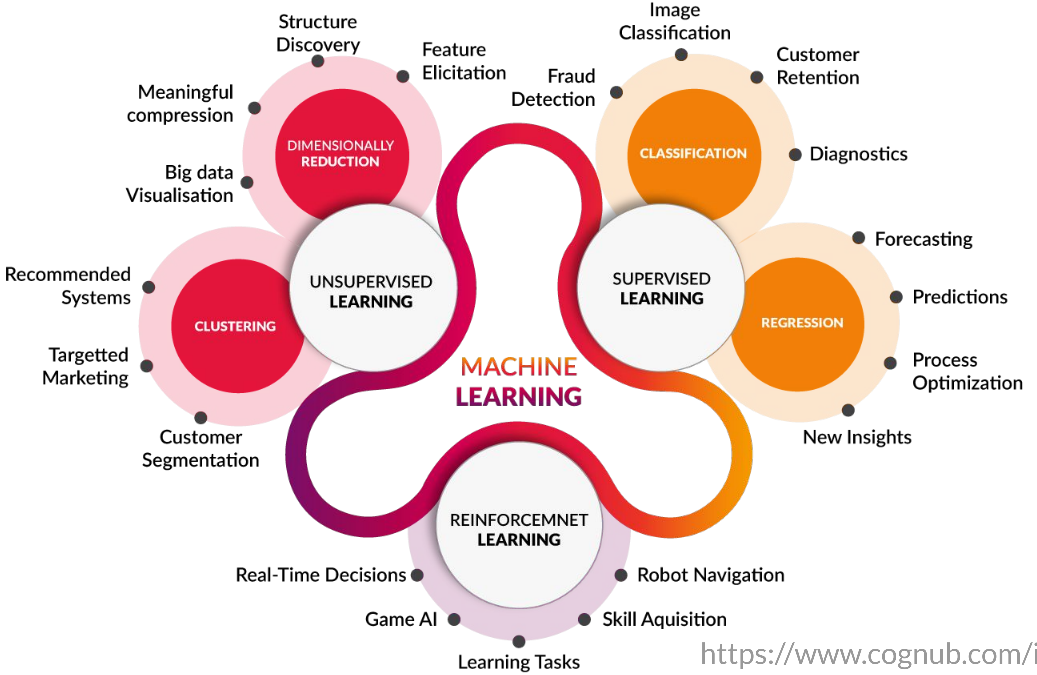
Reinforcement Learning 101

Kim Alyona

Summary

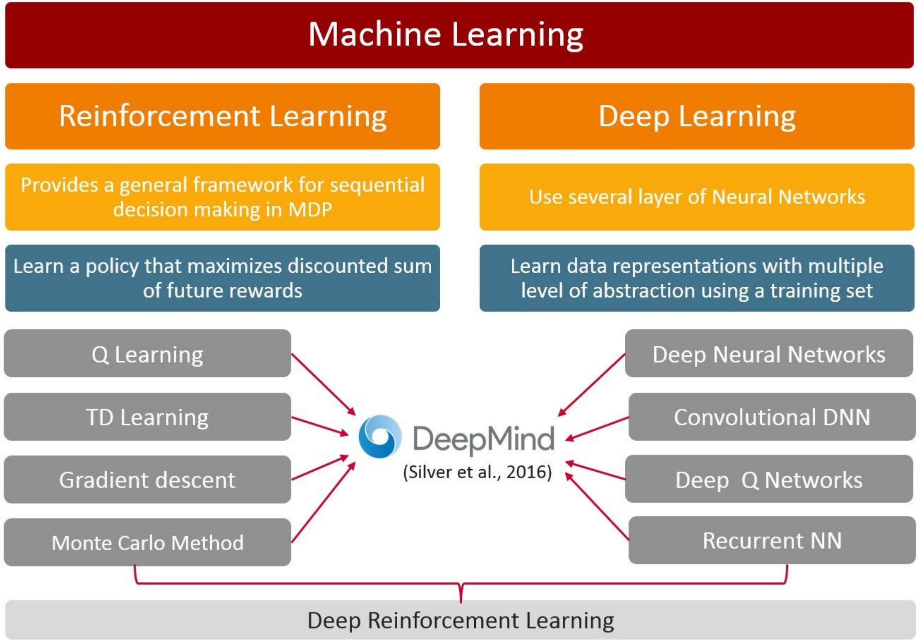
1. Introduction
2. Key concepts
3. Approaches
    a. common
    b. extra
4. Known problems
5. Applications

What is Reinforcement Learning?



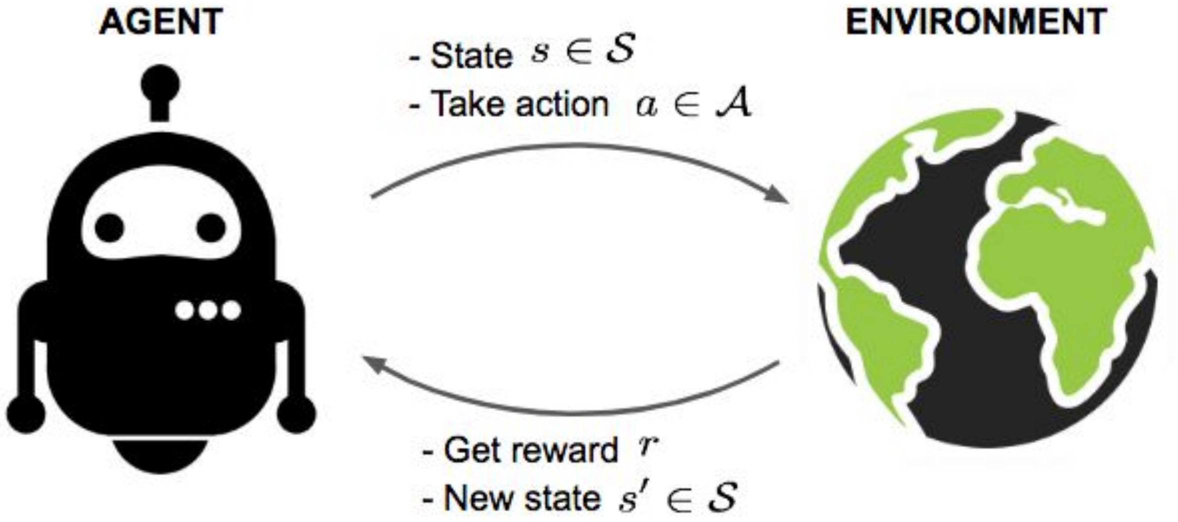
https://www.cognub.com/index.php/cognitive-platform/

Reinforcement vs Deep



https://towardsdatascience.com/from-classic-ai-techniques-to-deep-learning-753d20cf8578

Key Concepts



https://lilianweng.github.io/lil-log/2018/02/19/a-long-peek-into-reinforcement-learning.html

Key Concepts

Policy $\pi(s)$ tries to maximize the sum of rewards. *Agent's "brain"*

- deterministic $\pi(s) = a$
- stochastic $\pi(a|s) = \mathbb{P}_\pi[A = a|S = s]$
  - different sampling procedures

Key Concepts

Return

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- discounted sum
- more uncertainty in the future
- future actions do not result in immediate benefits
- math convenience

https://lilianweng.github.io/lil-log/2018/02/19/a-long-peek-into-reinforcement-learning.html

Key Concepts

State-value function $V_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$

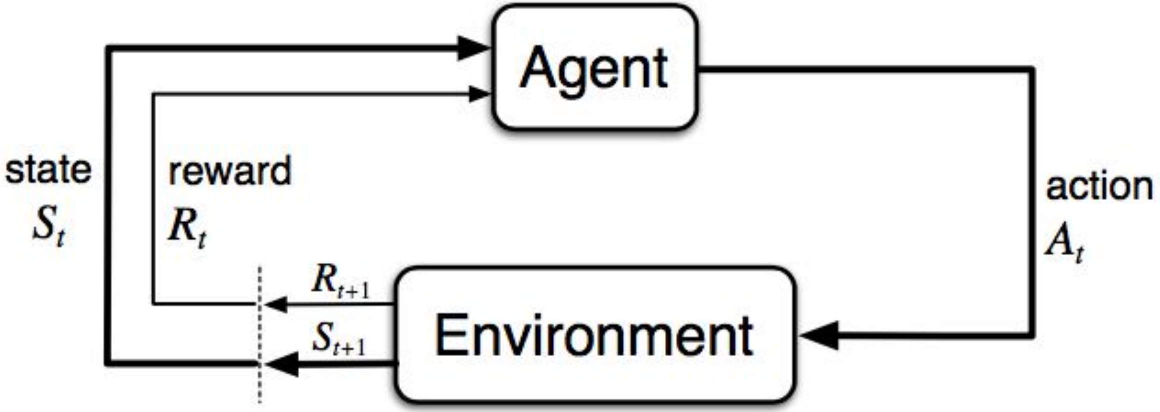
Action-value function $Q_\pi(s,a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$

math. property $V_\pi(s) = \sum_{a \in A} Q_\pi(s,a)\pi(a|s)$

Advantage $A(s,a) = Q_\pi(s,a) - V_\pi(s)$

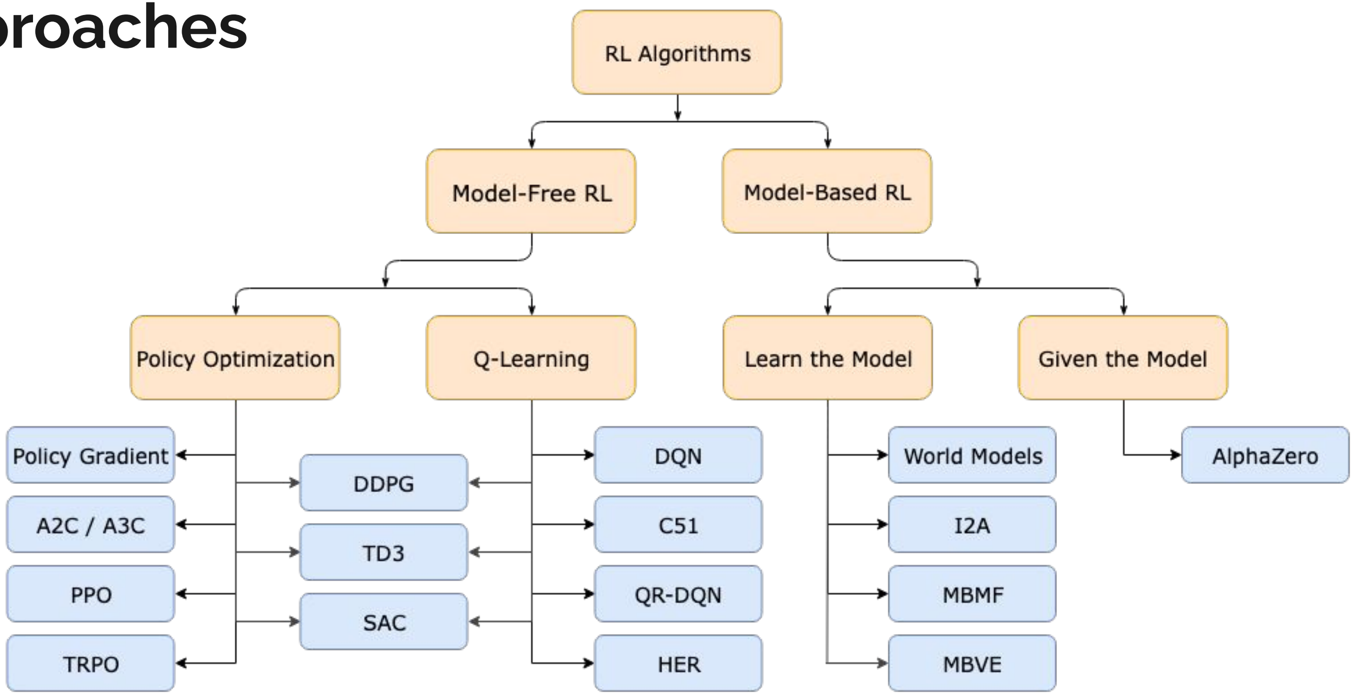
Markov Decision Process

$$\mathbb{P}[S_{t+1} | S_t] = \mathbb{P}[S_{t+1} | S_1, \ldots, S_t]$$



Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction; 2nd Edition. 2017.

Approaches



https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html

Approaches

Model-based: needs environment. States & Rewards are known or learnt

Model-free: does not depend on environment model during training

On-policy: use the deterministic **outcomes** *or* samples from the target policy

Off-policy: training on a distribution of **transitions** or episodes produced by a different behavior policy rather than that produced by the target policy.

Policy Optimization

- model-free & on-policy
- represent a policy explicitly $\pi_\theta(a|s)$
- usually involves learning an approximator for the value-function

Example:

- Asynchronous Advantage Actor-Critic (A3C)
- **Critic**: updates value function *V(s; w)* parameters *w*.
- **Multiple Actors**: updates policy parameters *θ*, in the direction suggested by the critic, π(a|s;θ)
- $J_V(w) = (G_t - V(s;w))^2$

Q-learning

- model-free & off-policy
- Bellman equations
- policy: $$a(s) = \arg\max_a Q_\theta(s, a)$$
- temporal difference

Example:

- Deep Q-Network (DQN)
    - Experience Replay
    - Periodically Updated Target

Trade-offs

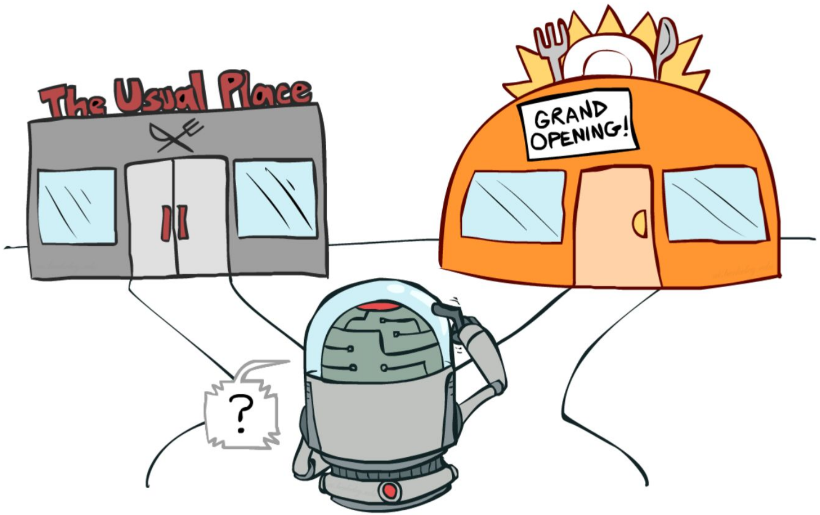
Policy Optimization	Q-learning
• you directly optimize for the thing you want (policy) • stable and reliable • less sample efficient	• indirectly optimize for agent performance (training to satisfy a self-consistency equation) • less stable • more sample efficient, can re-use data

Model-based RL

- Background: Pure Planning (MBMF)
  - model-predictive control
  - never explicitly represents policy (an optimal plan over fixed time window)
- Expert Iteration (ExIt, AlphaZero)
  - planning algorithm like Monte-Carlo Tree Search
  - sampling from the current policy & evaluate samples with planning algorithm
- Data Augmentation for Model-Free Methods (MBVE)
  - augment real data with synthetic

Known Problems

- Exploration-Exploitation Dilemma
- Deadly Triad Issue
  - off-policy
  - nonlinear function approximation
  - bootstrapping
  - **unstable learning, does not converge**

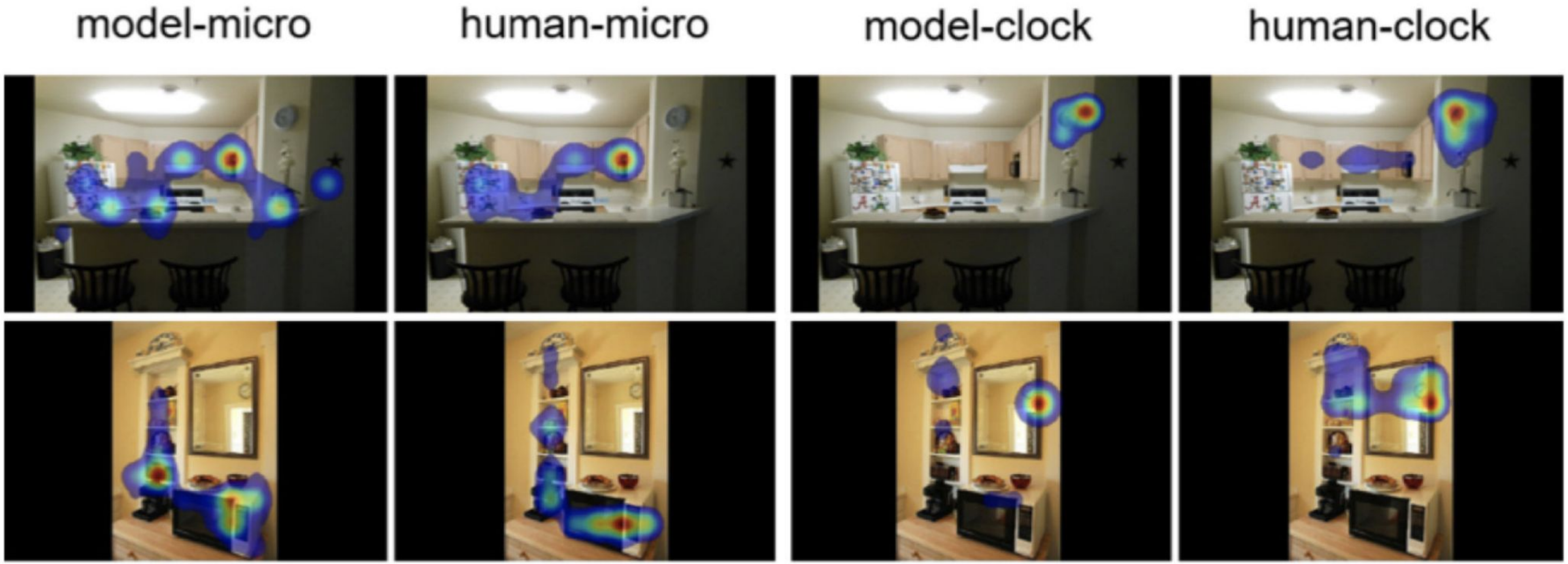


http://ai.berkeley.edu/lecture_slides.html, lecture 11

Applications

- modeling and explaining neural activity
- linking phasic **dopamine release** with temporal-difference **reward-prediction error** [Niv 19]
- learning complex **robotic skills** from raw **sensory input**
- modeling goal-oriented search **policy** with **top-down factors** as **states**

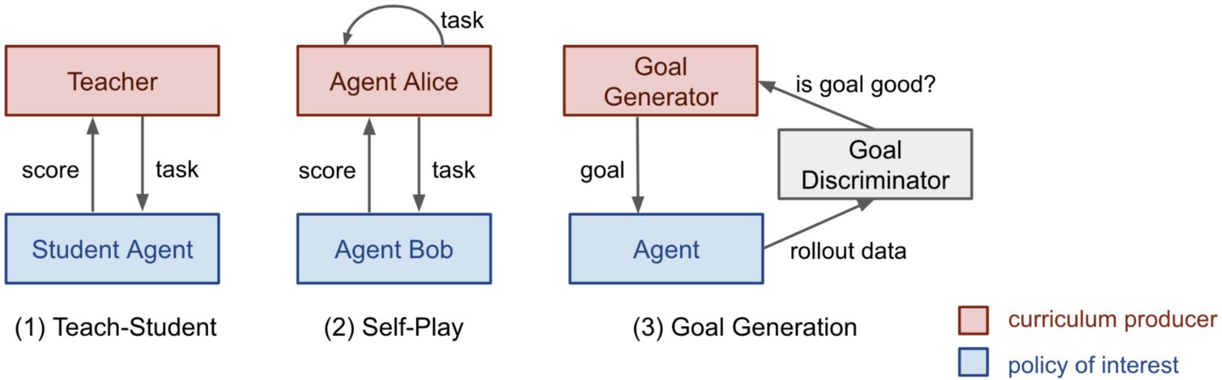
Spoiler for my next talk



Chen, Y., & Zelinsky, G. (2020). Predicting Goal-directed Attention Control Using Inverse Reinforcement Learning and COCO-Search18. *Journal of Vision*, *20*(11), 1632-1632.

Extra slide

- Curriculum RL
- Meta-Reinforce Learning
- Inverse RL



https://lilianweng.github.io/lil-log/2020/01/29/curriculum-for-reinforcement-learning.html