Reinforcement Learning

Extended RL, AGI and applications

Mario Martin

CS-UPC

May 20, 2021

Interesting topics not discussed in the course

- There are still some hot topics we haven't covered, some of them related to AGI and Long-life RL:
  - Hierarchical reinforcement learning
  - Transfer Learning
  - Meta-Learning
  - Links with neuro-physiology
- Today, we'll mention some of them
- We will also see some successful applications of RL

Extended RL: towards AGI

AI and RL

- RL is well suited to the idea of AI agent
- Learns a behavior to fulfill own goals
- Is grounded in the environment
- Has the notion of optimality but with given resources (rationality)

AI and RL

- But still some problems for true AI agents
- Most important is that learning is only of one behavior defined by reward function
- When learning another task, learning has to start from scratch
- Too many interactions with env. for learning
- Not suited to the idea of long-live learning
- Some steps in solving these limitations
  - Transfer of learning
  - Multi-task learning
  - (Curriculum learning)
  - Hierarchical Learning
  - Meta-learning

Transfer RL

- Can we extend knowledge generated in one task to a different task?
- Changes in the task: different dynamics, different reward and/or different actions.
- Several ways to do that:
  - Learn one task and start with policy and values and finetune to next task
  - Randomization of the input to prepare for other scenarios or use of entropy in the policy
  - MultiTask and Meta-learning (see next slides)
  - Transfer of info from one task to the other (Q-values, policy, reward, samples, model, features, etc.)
- Example: Sharing of examples and IRL for task disentangled from actions (AIRL)
- See recent survey

Hierarchic RL

- Also helps in sparse rewards, but also useful for transfer learning.
- Natural way of learning.
- In some cases a complex task can be decomposed in simpler tasks.
- Learning is simplified when first these tasks are learnt.
- Several ways to find that:
  1. Using subrewards for subactions (reward shaping)
  2. **Discover them automatically**
- Actions can be reused to learn other tasks
- See references about the topic in course web page

Multi-task learning

- What is a task?

$$\text{A task:} \quad \mathcal{T}_i \triangleq \left\{\mathcal{S}_i, \mathcal{A}_i, p_i\left(s_1\right), p_i\left(s' \mid s, a\right), r_i(s, a)\right\}$$

- Changes in on item means a different task
- Agent does not only solve one task but several

Multi-task learning

- What is a task?

$$\text{A task:} \quad \mathcal{T}_i \triangleq \left\{\mathcal{S}_i, \mathcal{A}_i, p_i\left(s_1\right), p_i\left(s' \mid s, a\right), r_i(s, a)\right\}$$

- Changes in on item means a different task
- Agent does not only solve one task but several

Multi-task RL

- Given: a set of training tasks
- Goal: Learn a policy that can solve different tasks

Multi-task learning

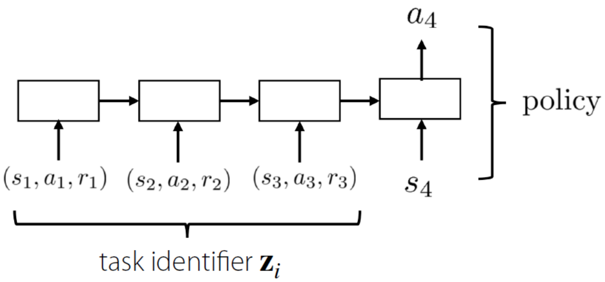
- Optimize learning/performance across all tasks through shared knowledge.
- Conditioned policies (see previous lecture) are a kind of Multi-task RL where each task is characterized by a goal state
- In Multi-task, we have conditioned policy again, but more general.

$$\pi(s, z)$$

where z is indicative of the task

Multi-task learning

- Hmmm. But indication of task is reward function or dynamics or starting state
- In some formulations the agent has to discover the scenario from the rewards he obtain
- This means use of memory:



Meta-Learning in RL

- Similar to multi-task learning but different focus and procedure

Meta RL

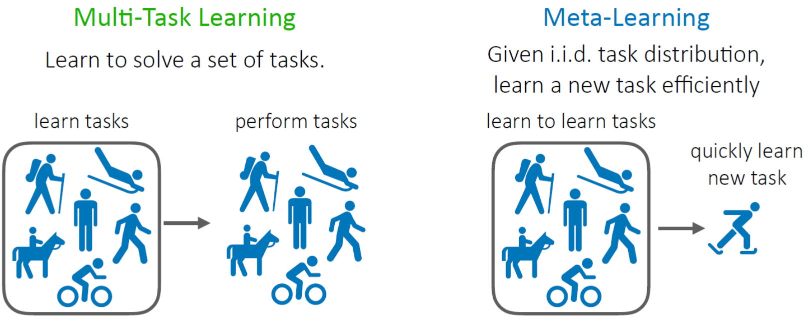
- Given: a set of training tasks
- Goal: Learn to solve those task and also can be learn *efficiently* new tasks

- Formulation: Given a set of training tasks, learn a policy that can also be applied successfully (directly or after small finetuning) to a set of testing tasks.

Meta-Learning in RL

- Procedure to learn is done as follows:
  1. Sample one task $\mathcal{T}_i$ from set of training tasks
  2. Generate N episodes for task $\mathcal{T}_i$ with policy
  3. Store data in ED for $\mathcal{T}_i$
  4. Update policy to maximize discounted return for all tasks.
- Focus on efficiently learn a set of different tasks.
- Learn-to-learn idea at the beginning, but extended also to generalization between tasks: Learning of each task has to be consistent and (hopefully) helpful for learning other tasks.
- Very popular in two last years, in ML and RL in particular (See course CS330 from Stanford here)
- See also a specific introduction and review of latest approaches for RL

Comparison meta and multi RL



Life long learning in RL

- Our agents may not be given a large batch of data/tasks right off the bat!
- Tasks are presented in sequence:

- Also called sequential learning, continual learning or incremental learning
- No forgetting part absent in Meta-RL

Links with neuro-physiology

- In Artificial Intelligence it is important to compare techniques with actual techniques used by humans and animals
- Reinforcement learning has its roots in intuitive idea of learning in animals.
- Lately, a lot of papers support that RL is implemented in the brain
  - Link of RL with actual learning in brain
  - Dopamine as reward
  - Dopamine implements TD error.
  - Support for dopamine acting as distributed value estimation

Some succesful aplications of RL

Applications

- Games: Backgammon, Go, Chess (competition), Star-Craft, Dota 2, Poker (Pluribus)
- Robotics: Walking, Manipulation (also here), etc.
- Medicine: Review. Example: Sepsis treatment, or ventilation
- Drug dessign: For instance here.
- Physics
- Recommender systems
- Finances
- Optimization in general, f.i control power, or for IoT,
- Mathematics: For instance, Logic profs
- Natural Language processing: Summarizing texts.