Part 1: The Idea of Reinforcement Learning

Learning from interaction
- with environment
- to achieve some goal

- Example 1 Baby playing: No teacher; sensorimotor connection to environment.
  - Cause-effect/Action-consequences
  - How to achieve some goal
- Example 2 Learning to hold a conversation, etc.
  - We find out the effects of our actions later.
Supervised Learning

Training Info = desired (target) outputs

Inputs  ➔ Supervised Learning System  ➔ Outputs

Error = (target output – actual output)
Reinforcement Learning

Training Info = evaluations ("rewards" / "penalties")

Inputs  

RL System  

Outputs ("actions")

Objective: get as much reward as possible
R L - How does it work?

Learning a mapping from situations to actions in order to maximize a scalar reward/reinforcement signal.

How?

- Try out actions to learn which produces highest reward - *trial and error search*
- Actions affect immediate reward + all subsequent rewards - *delayed effects, delayed rewards*
Exploration/Exploitation Trade-off

- High rewards from trying previously-well-rewarded actions - EXPLOITATION (= greedy)
- BUT: Which actions are best? Must try ones not tried before - EXPLORATION (= $\varepsilon$)

Must do both!

- Exploitation/Exploration trade-off also depends on the life-time of an agent.
Part 2: Framework of RL

- Temporally situated
- Continual learning and planning
- Object is to *affect* the environment
- Environment is stochastic and uncertain
Elements of RL

- **Policy**: what to do
- **Reward**: what is good
- **Value**: what is good because it *predicts* reward
- **Model**: what follows what
General RL Algorithm

i. Initialise learner’s internal state

ii. Do forever (!?):
   1. Observe current state $s$
   2. Choose action $a$ using some evaluation function
   3. Execute action $a$
   4. Let $r$ be immediate reward, $s’$ new state
   5. Update internal state based on $s,a,r,s’$
Summary: Key Features of RL

✓ Learner is not told which actions to take
✓ Trial-and-Error search
✓ Possibility of delayed reward
  (Sacrifice short-term gains for greater long-term gains)
✓ The need to *explore* and *exploit*
✓ Considers the whole problem of a goal-directed agent interacting with an uncertain environment