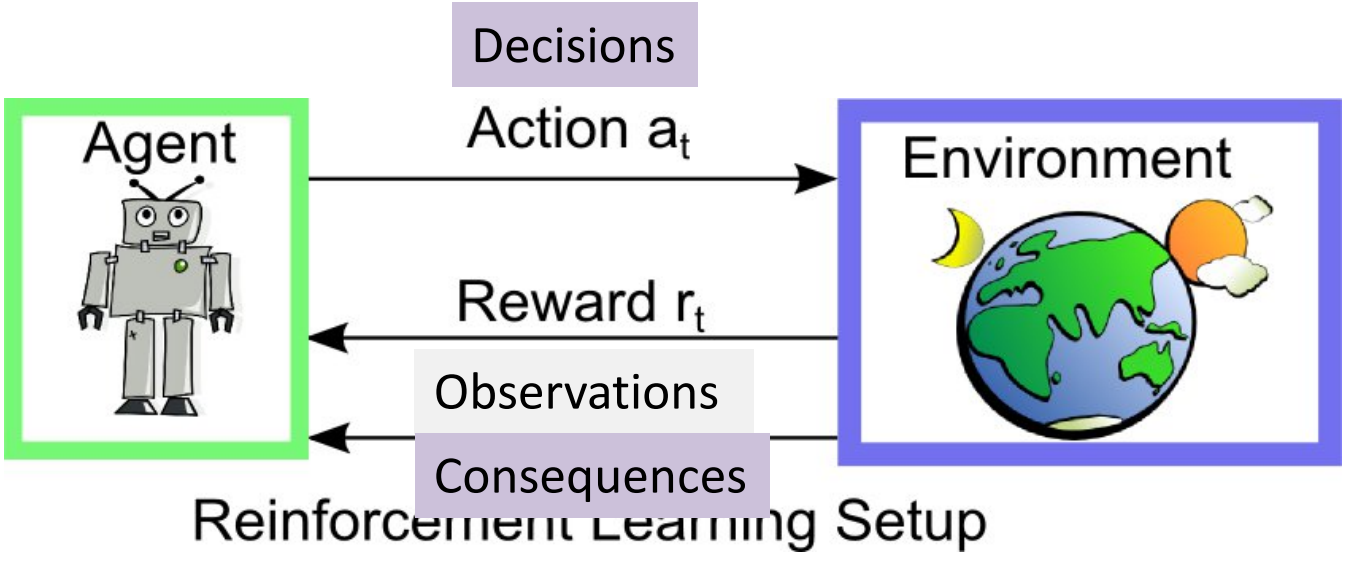
Deep RL

Mar 5 2018

Sudeshna Sarkar

RL


Decisions
Agent
Action a_t
Environment
Reward r_t
Observations
Consequences
Reinforcement Learning Setup

Examples

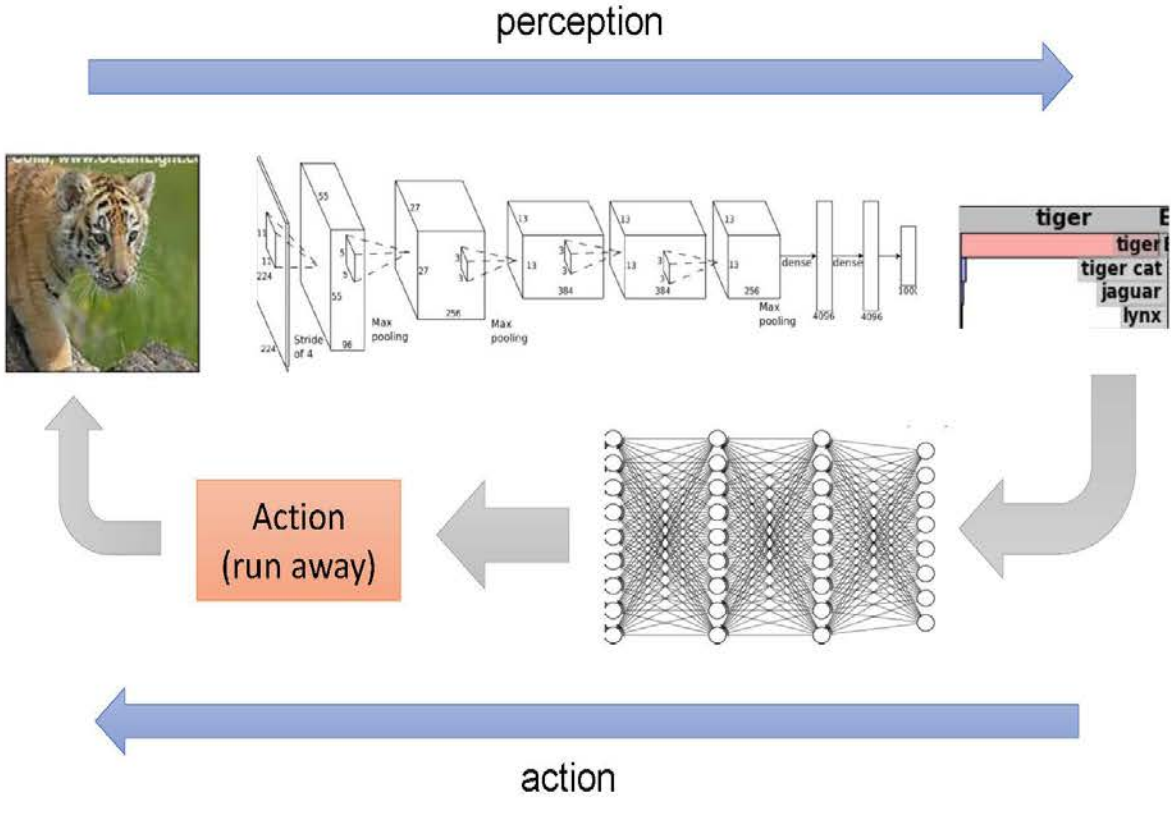
- Fly stunt manoeuvres in a helicopter
- Defeat the world champion at Backgammon
- Manage an investment portfolio
- Control a power station
- Make a humanoid robot walk
- Play many different Atari games better than humans

Characteristics of Reinforcement Learning

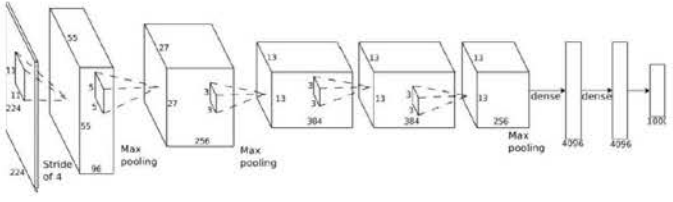
- There is no supervisor, only a reward signal
- Feedback is delayed, not instantaneous
- Time really matters (sequential, non i.i.d data)
- Agent's actions affect the subsequent data it receives

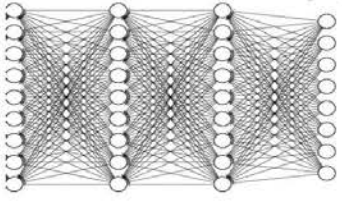
Deep RL

- DL : end to end training of expressive multi-layer models
- Use DL to allow RL algorithms to solve complex problems end to end!


perception
Stride of 4
Max pooling
Max pooling
Max pooling
dense
dense
tiger
tiger cat
jaguar
lynx
Action (run away)
action

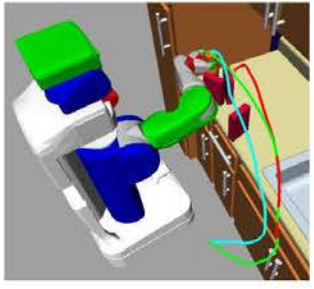
sensorimotor loop





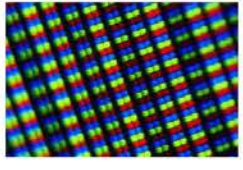
Action
(run away)

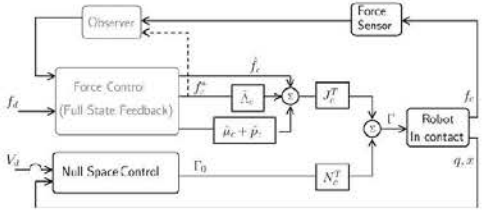
Example: robotics



robotic control pipeline

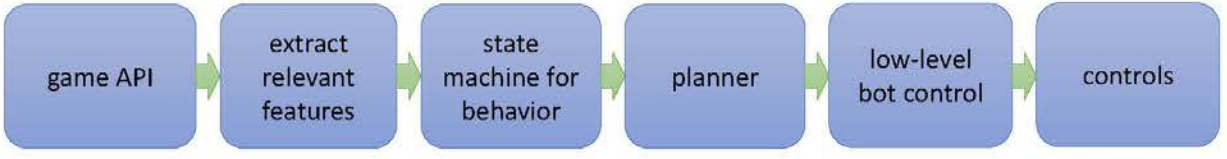
observations → state estimation (e.g. vision) → modeling & prediction → planning → low-level control → controls

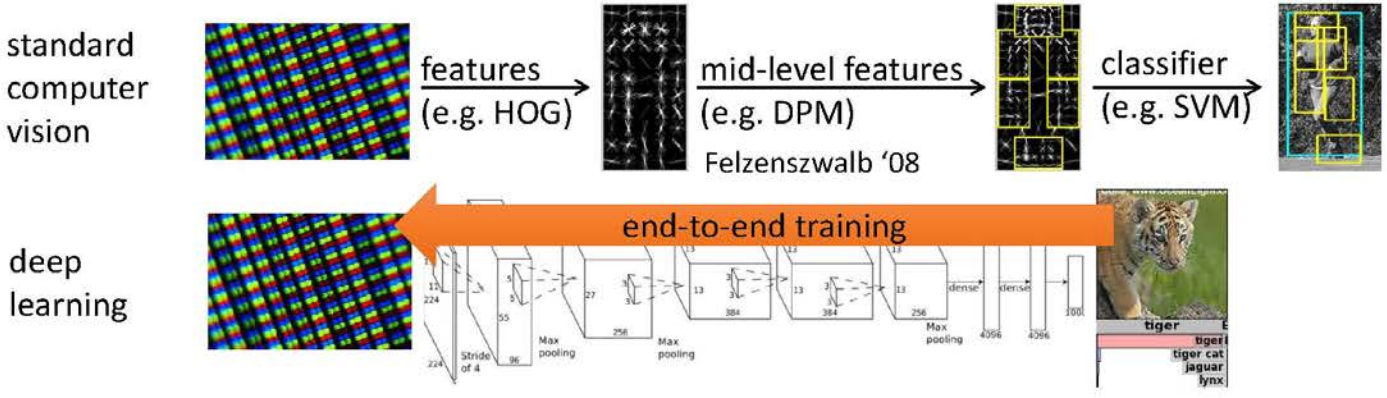


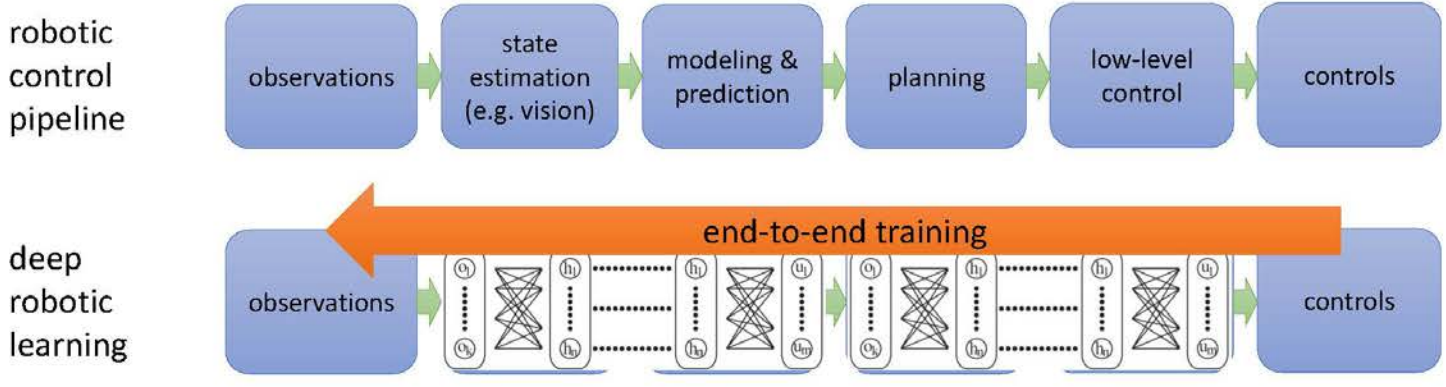


Example: playing video games

video
game
AI pipeline




standard computer vision
features (e.g. HOG)
mid-level features (e.g. DPM)
Felzenszwalb '08
classifier (e.g. SVM)
deep learning
end-to-end training
tiger
tiger cat
jaguar
lynx


robotic control pipeline
observations
state estimation (e.g. vision)
modeling & prediction
planning
low-level control
controls
deep robotic learning
end-to-end training
observations
controls

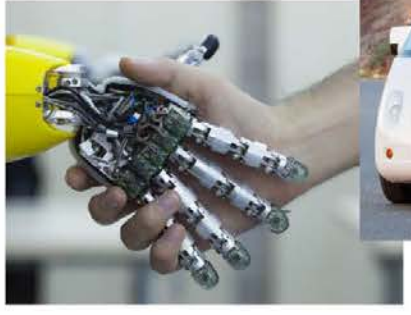
When to “sequential decision making”?

- Limited supervision
- Actions have consequences

Common Applications

autonomous driving

business operations



robotics

language & dialogue
(structured prediction)

finance

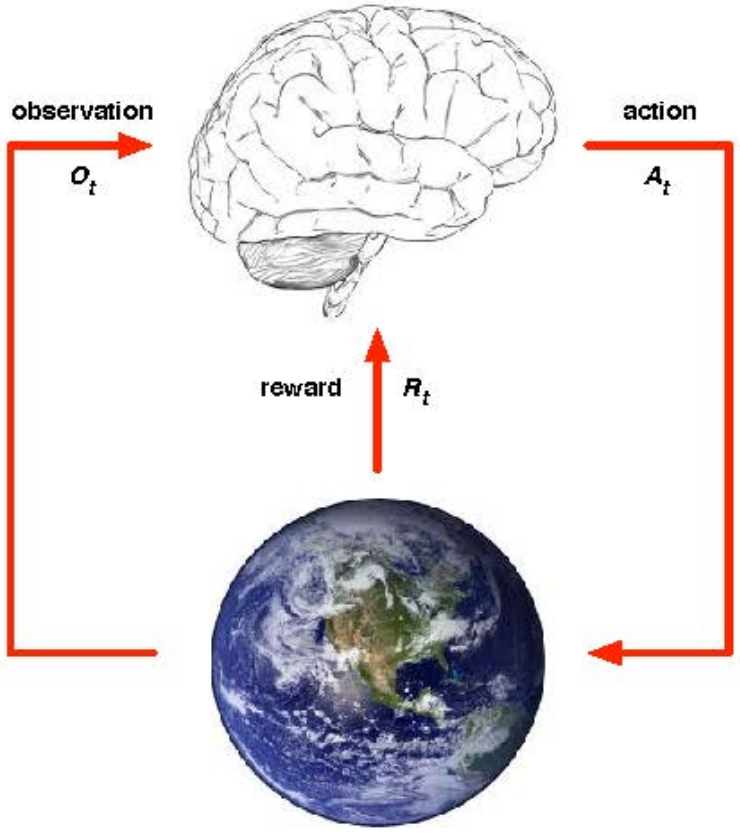
Beyond Learning from reward

- Basic RL deals with maximizing rewards
- This is not the only problem that matters for sequential decision making!

1. Learning reward functions from example (inverse RL)
2. Transferring skills between domains
3. Learning to predict and using prediction to act

- Imitation Learning: supervised learning for decision making
  - Does direct imitation work?
  - How can we make it work more often?
- Inferring Intentions

Agent and Environment



- At each step t the agent:
  - Executes action A_t
  - Receives observation O_t
  - Receives scalar reward R_t
- The environment:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}
- t increments at env. step

History and State

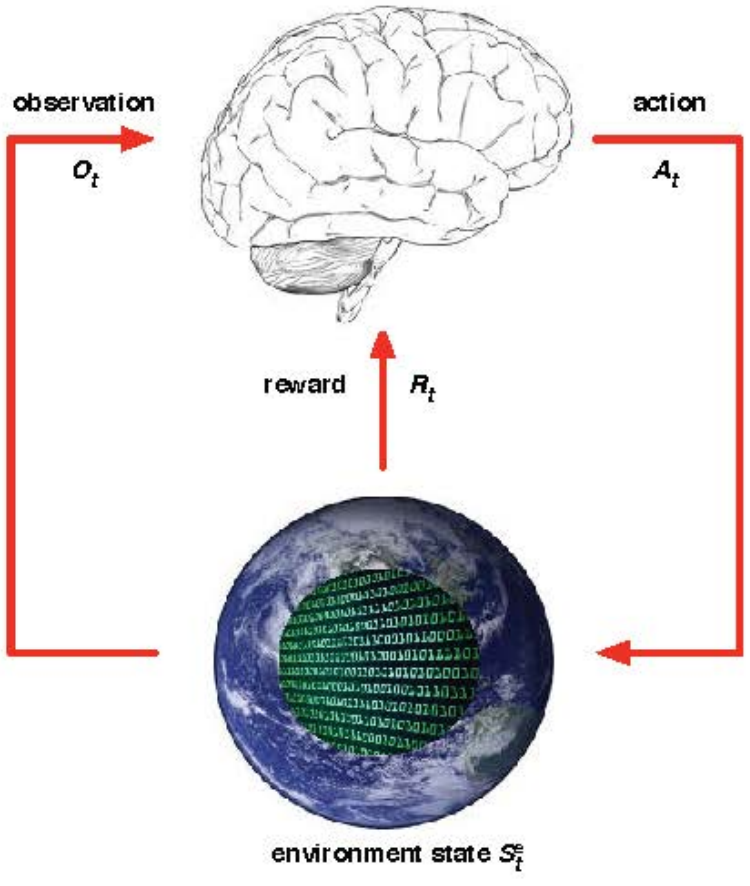
- The history is the sequence of observations, actions, rewards

$$H_t = O_1, R_1, A_1, ..., A_{t-1}, O_t, R_t$$

- i.e. all observable variables up to time t
- i.e. the sensorimotor stream of a robot or embodied agent
- What happens next depends on the history:
  - The agent selects actions
  - The environment selects observations/rewards
- State is the information used to determine what happens next
- Formally, state is a function of the history:

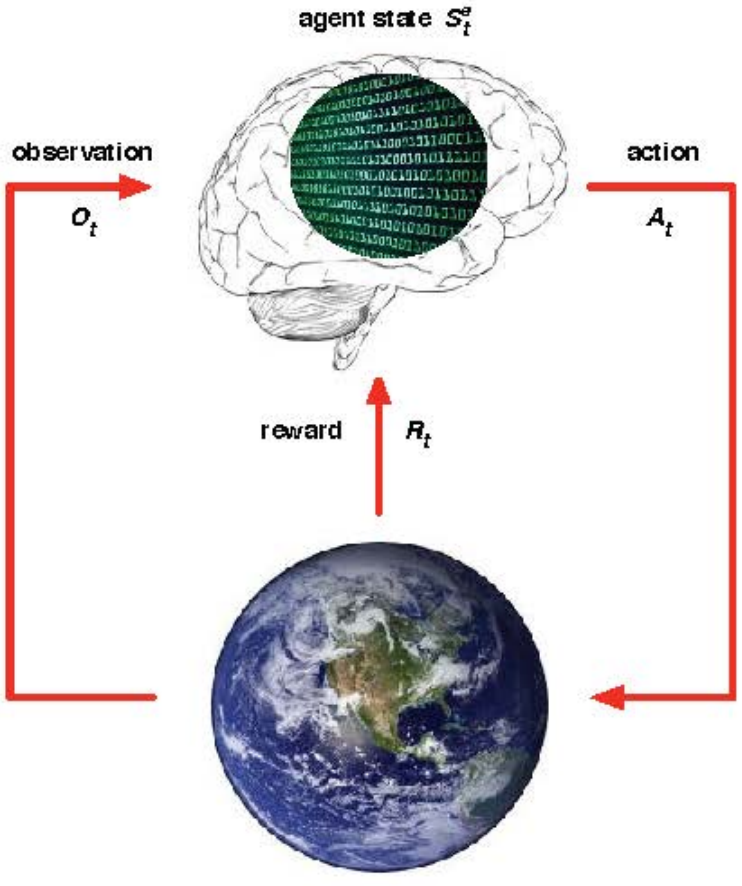
$$S_t = f(H_t)$$

Environment State



- The environment state S_t^e is the environment's private representation
- i.e. whatever data the environment uses to pick the next observation/reward
- The environment state is not usually visible to the agent
- Even if S_t^e is visible, it may contain irrelevant information

Agent State



- The agent state S_t^a is the agent's internal representation
- i.e. whatever information the agent uses to pick the next action
- i.e. it is the information used by reinforcement learning algorithms
- It can be any function of history:

$$S_t^a = f(H_t)$$