Reinforcement Learning

Lecture 9: Model-based methods

Chris G. Willcocks

Durham University

Lecture overview

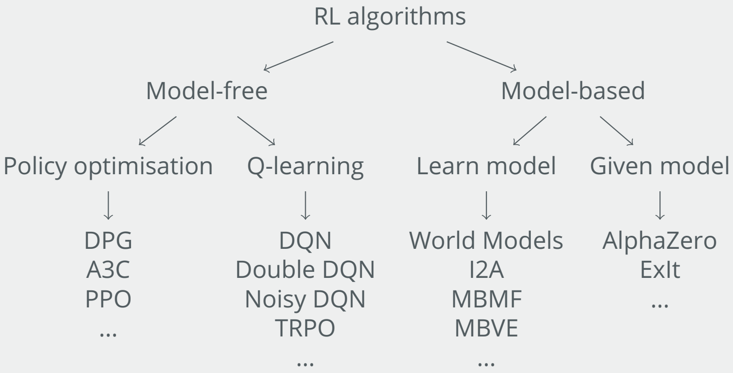
Lecture covers chapter 8 in Sutton & Barto [1] and examples from David Silver [2]

1. **Model-based reinforcement learning**
   - taxonomy
   - overview
   - the simulation cycle
   - characteristics

2. **Integrated learning and planning**
   - Dyna-Q
   - characteristics
   - Monte Carlo tree search
   - simulated policy learning

Model-based RL taxonomy

Taxonomy of reinforcement learning algorithms



RL algorithms

Model-free

Model-based

Policy optimisation

Q-learning

Learn model

Given model

DPG
A3C
PPO
...

DQN
Double DQN
Noisy DQN
TRPO
...

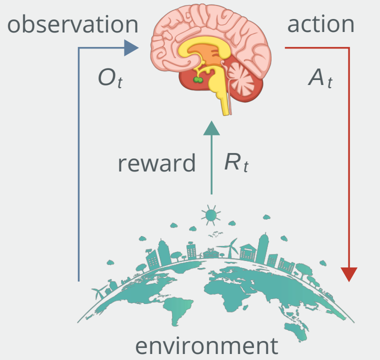
World Models
I2A
MBMF
MBVE
...

AlphaZero
ExIt
...

This figure does not capture overlap, for example between policy optimsiation and Q-learning algorithms

Model-based RL overview

RL Agents



observation

O_t

action

A_t

reward

R_t

environment

In **model-free** RL:

No model

Learn the value function $q(s; a)$ and/or the policy (ajs) from experience

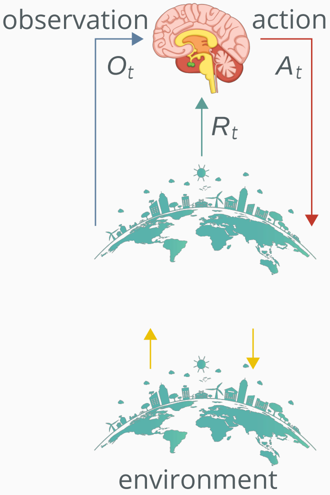
In **model-based** RL:

Learn the model from experience

Plan the value function and/or the policy from the model

Figure based on [2, 1]

Model-based RL the simulation cycle



Model-based RL cycle:

The agent experiences the real environment

We learn a model to predict what the real environment does (when you take an action)

We then use this simulated model to plan

This allows us to estimate the value function and/or policy without directly interacting with the real environment

But we use this policy to take real actions again

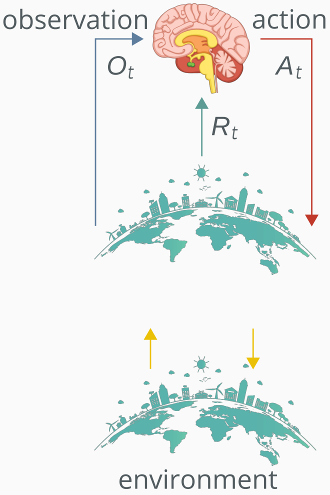
Model-based RL characteristics

Model-based RL advantages:

- The model can sometimes be a simpler and more useful representation of the environment than you can otherwise access by experience
- Can be learnt by supervised learning
- Can reason about model uncertainty

Model-based RL disadvantages:

- This is another component which introduces some approximation error
  - Value function and/or policy approximation and now model approximation
- We can only be as good as our model



Model-based RL definition

Definition: model

A model $M \ = hP \ \ ; R \ \ i$ is a parameterised representation of an MDP: $hS; A; P; R i$. It approximates state transitions $P \qquad P$ and rewards $R \qquad R$, learning a distribution over the next states and rewards:

$$\begin{aligned} S_{t+1} & \quad P(S_{t+1} j S_t; A_t) \\ R_{t+1} & = R(R_{t+1} j S_t; A_t); \end{aligned}$$

which typically are conditionally independent of each other:

$$\begin{aligned} & P(S_{t+1}; R_{t+1} j S_t; A_t) = \\ & \qquad P(S_{t+1} j S_t; A_t) P(R_{t+1} j S_t; A_t) \end{aligned}$$

Example: environment model

Model-based RL learning the model

Learning the model

We learn the model M from experience $\{S_1, A_1, R_2, \ldots, S_T\}$ using **supervised learning**.

We receive a stream of actual experiences

This gives us a dataset:

$$\begin{aligned} S_1, A_1 &\to R_2, S_2 \\ S_2, A_2 &\to R_3, S_3 \\ &\vdots \end{aligned}$$

$s, a \to r$ is a regression problem

$s, a \to s'$ is a density estimation problem

Integrated learning simulated and real experience

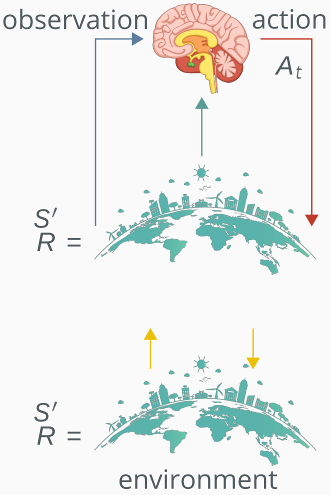
Experience can be simulated and real

Simulated experience sampled from M

$$S^{0} \quad P\ (S^{0}jS;A)$$
$$R = R\ (RjS;A)$$

Real experience sampled from the true MDP

$$S^{0} \quad P^{a}_{s;s'}$$
$$R = R^{a}_{s}$$

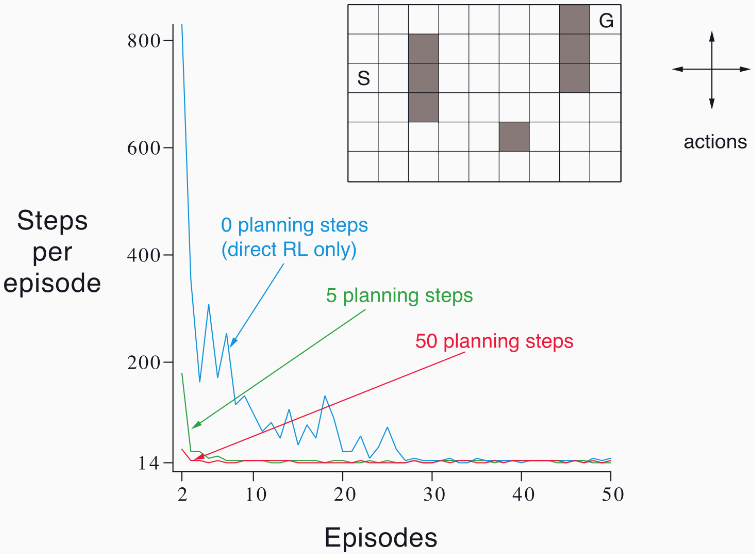


Model-based RL Dyna-Q

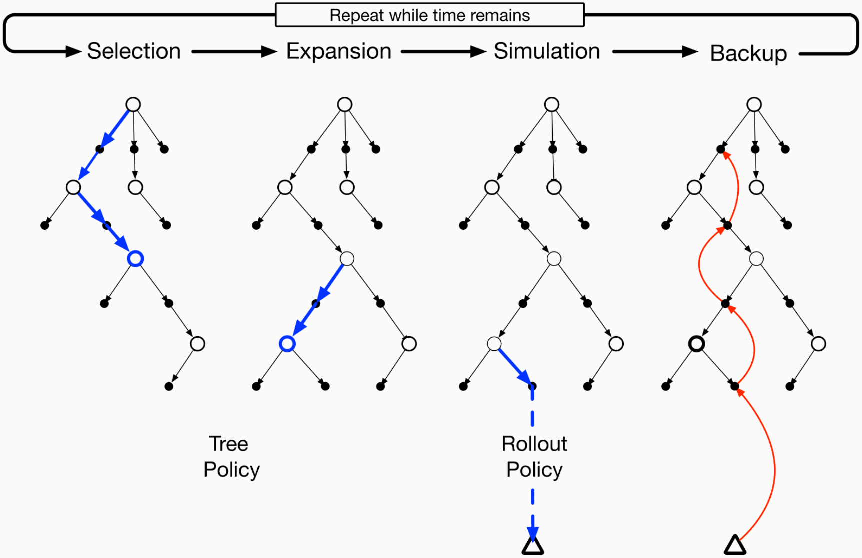
Algorithm: Dyna-Q [3, 4]

```
initialise $Q(s; a)$ and model $M(s; a)$ for all $s \; 2 \; S$ and $a \; 2 \; A(s)$
while True:
  $s$   current (nonterminal) state
  $a$     greedy$(s; Q)$
  $r; s^0$    env.step$(s; a)$
  $Q(s; a)$    $Q(s; a) + \;\; (r + \;\; \max_{\hat{a}} Q(s^0; \hat{a}) \;\; Q(s; a))$
  $M(s; a)$    $r; s^0$ (assuming deterministic environment)
  for i in range(n):
    $s$   random previously observed state
    $a$   random action previously taken in $s$
    $r; s^0$    $M(s; a)$
    $Q(s; a)$    $Q(s; a) + \;\; (r + \;\; \max_{\hat{a}} Q(s^0; \hat{a}) \;\; Q(s; a))$
```

Model-based RL Dyna-Q characteristics

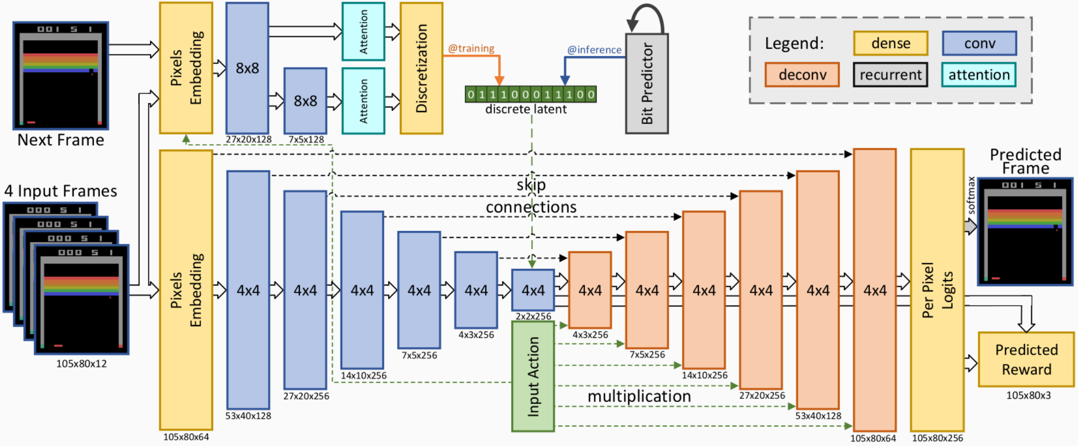


Model-based RL Monte Carlo tree search (MCTS)



Model-based RL simulated policy learning

“Model-Based Reinforcement Learning for Atari” [5]



Take Away Points

Summary

In summary, model-based methods:

- are easy to train with supervised learning
- allow for planning ahead
- can be very data efficient
- can be used to imagine situations without experiencing them
- but the value and policy learnt can only be as good as the model
- they can be combined with model-free methods

References I

[1] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction (second edition). Available online. MIT press, 2018.

[2] David Silver. Reinforcement Learning lectures. https://www.davidsilver.uk/teaching/. 2015.

[3] Richard S Sutton. “Integrated architectures for learning, planning, and reacting based on approximating dynamic programming”. In: Machine learning proceedings 1990. Elsevier, 1990, pp. 216–224.

[4] Baolin Peng et al. “Deep Dyna-Q: Integrating planning for task-completion dialogue policy learning”. In: arXiv preprint arXiv:1801.06176 (2018).

[5] Lukasz Kaiser et al. “Model-based reinforcement learning for atari”. In: arXiv preprint arXiv:1903.00374 (2019).