Reinforcement Learning: 1. Introduction

Lecturer: Michael Herrmann
IPAB, School of Informatics
michael.herrmann@ed.ac.uk
Informatics Forum 1.42, 651 7177

12/01/2016

Admin: Time split

- Lectures ($\leq$20h): Tuesday and Friday 12:10 - 13:00 (Teviot LT, Medical School, Gateway 5)
- Assessment: Homework and Exam = 10% + 10% and 80%
- HW1 (10h): out by 21 January, due 11 February (possible topic: $\mathcal{Q}$-learning: A learning agent in a box-world)
- HW2 (10h): out by 25 February, due 17 March (possible topic: Continuous-space RL)
- Reading/SelfStudy/Preparation of tutorial problems ($\approx$30h)
- Tutorials (8h) starting in week 3
- Revision (20h)

Admin: Tutorials

- Mondays, Thursdays or Fridays, 13:10 pm - 14:00 pm in **FH 1.B32**
- Topics, tentatively:
  - T1 [Bandit problems] – week of 25th Jan
  - T2 [Q-learning] – week of 1nd Feb
  - T3 [MC methods] – week of 8th Feb
  - T4 [TD methods] – week of 22th Feb
  - T5 [POMDP] – week of 29th Feb
  - T6 [continuous RL] – week 7th of Feb
  - T7 [practical aspects] – week 14 of Mar
  - T8 [revision] – week of 21th Mar
- We'll assign questions (combination of pen&paper and computational exercises – you attempt them before sessions) and give an opportunity to run some simulations
- Tutor will discuss and clarify concepts underlying exercises
- Tutorials are not assessed; gain feedback from participation

Admin: Resources

Webpage: www.informatics.ed.ac.uk/teaching/courses/rl

Lecture slides will be uploaded as they become available

Readings:

1. R. Sutton and A. Barto, Reinforcement Learning, MIT Press, 1998, webdocs.cs.ualberta.ca/~sutton/book/the-book.html
2. Csaba Szepesvari: Algorithms for Reinforcement Learning, Morgan & Claypool, 2010.
3. S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, MIT Press, 2006 (Chapters 14 – 16)
4. M. Wiering and M. v. Otterlo (eds.) Reinforcement Learning: State-of-the-art. Vol. 12. Springer, 2012.
5. Reinforcement Learning Warehouse http://reinforcementlearning.ai-depot.com/
6. Wikipedia (see also external links), Scholarpedia
7. Research papers (later)

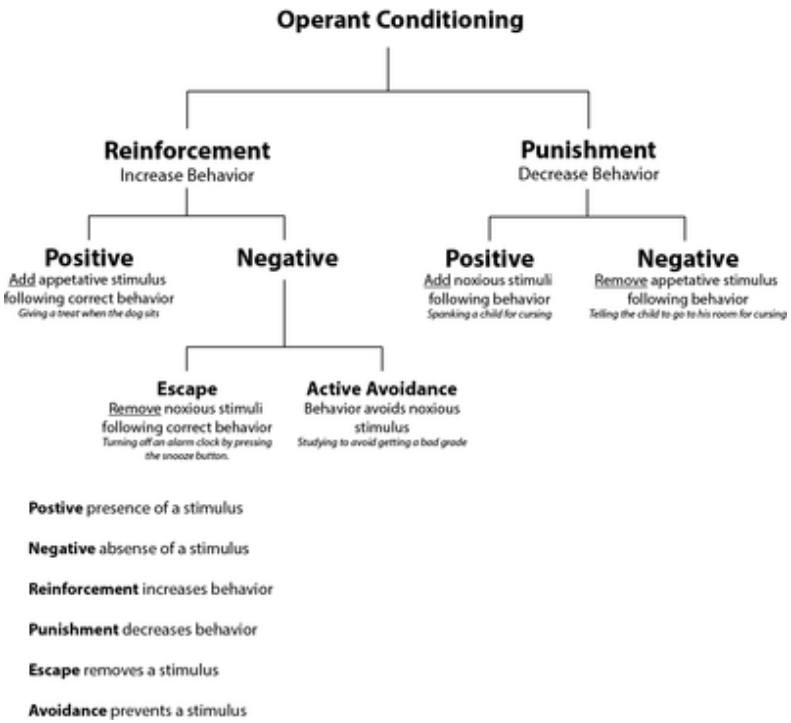
Background: mathematics, statistics, machine learning, matlab, ...

Reinforcement Learning

I. Psychology: Operant conditioning

Example: Whenever a rat presses a button, it gets a treat. If the rat starts pressing the button more often, the treat serves to positively reinforce this behaviour.

Example: Whenever a rat moves to a certain area of the cage, it gets a mild electric shock. If the rat starts moving to this area less frequently, the shock serves as a (positive) punishment.



en.wikipedia.org/wiki/Operant_conditioning

en.wikipedia.org/wiki/Reinforcement

Example: Shaping

Shaping is reinforcement of successive approximations to a desired instrumental response. In training a rat to press a lever, for example, simply turning toward the lever is reinforced at first. Then, only turning and stepping toward it is reinforced. The outcomes of one set of behaviours starts the shaping process for the next set of behaviours, and the outcomes of that set prepares the shaping process for the next set, and so on. As training progresses, the response reinforced becomes progressively more like the desired behaviour; each subsequent behaviour becomes a closer approximation of the final behaviour.

en.wikipedia.org/wiki/Reinforcement#Shaping

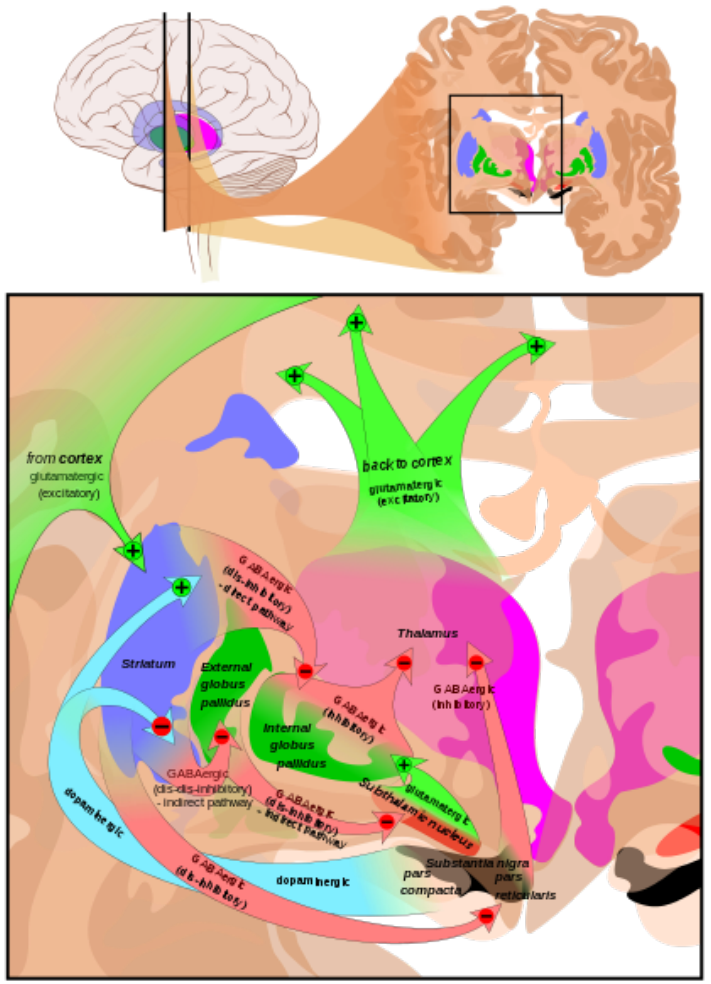
II. Brain science: Dopamine

An "important effect of dopamine is as a 'teaching' signal. When a motor response is followed by an increase in dopamine activity, the basal ganglia circuit is altered in a way that makes the same response easier to evoke when similar situations arise in the future. This is a form of operant conditioning, in which dopamine plays the role of a reward signal."

en.wikipedia.org/wiki/Dopamine#The_substantia_nigra_dopamine_system_and_motor_control

What can we learn from RL in the brain?

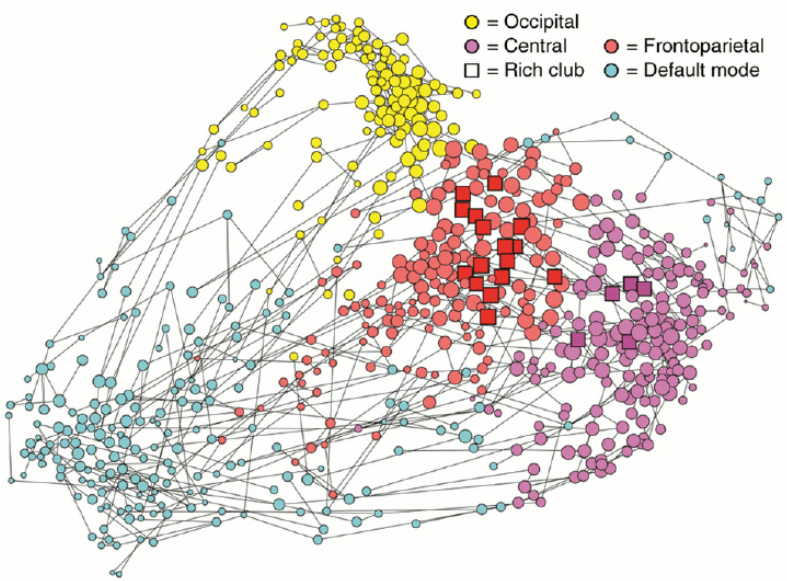
- A system that solves complex problems may have to be complex itself.
- "Until recently, rather little has been done to find out how animals behave, whether in the wild or captivity, when they have nothing particular to do." D. E. Berlyne, Conflict, Arousal, and Curiosity, McGraw-Hill, 1960. ⇒ Intrinsically-motivated RL (A. Barto)
- Time course of the reward and value processes



M. Häggström, A. Gillies and P. J. Lynch

What can we learn from RL in the brain?

- A system that solves complex problems may have to be complex itself.
- "Until recently, rather little has been done to find out how animals behave, whether in the wild or captivity, when they have nothing particular to do." D. E. Berlyne, Conflict, Arousal, and Curiosity, McGraw-Hill, 1960.
  $\Rightarrow$ Intrinsically-motivated RL (A. Barto)
- Time course of the reward and value processes



Functional Connectivity in the Brain

O. Sporns, Nature Neuroscience (2014)

III. Applications in Management, Education, Social science

B. F. Skinner's radical behaviourism: All learning results from reinforcement by repeated expected consequences.

Reactions (some random quotes)

- "The concept and use of rewards has been misunderstood for centuries!" (R. H. Richardson)
- "Negative reinforcement has its place in good management as well." (Neil Kokemuller)
- "If a leader chooses to rely heavily on rewards and punishments to meet his or her objectives, the leader must: determine what is "good" and "bad," ... Although that might be possible, the best outcome that could be hoped for would be mediocrity." (William Stinnett)
- "How to discipline without stress, punishment or rewards." "Education is about motivation." (Marvin Marshall)
- "You can't motivate the unmotivated." (Dan Gould)

III. Applications in Management, Education, Social science

B. F. Skinner's radical behaviourism: All learning results from reinforcement by repeated expected consequences.

Reactions (some random quotes)

- "The concept and use of rewards has been misunderstood for centuries!" (R. H. Richardson)
- "Negative reinforcement has its place in good management as well." (Neil Kokemuller)
- "If a leader chooses to rely heavily on rewards and punishments to meet his or her objectives, the leader must: determine what is "good" and "bad," ... Although that might be possible, the best outcome that could be hoped for would be mediocrity." (William Stinnett)
- "How to discipline without stress, punishment or rewards." "Education is about motivation." (Marvin Marshall)
- "You can't motivate the unmotivated." (Dan Gould)

Conclusion: Start with simple questions and simple systems.

What means "RL" (in Machine Learning)?

- Learning given only percepts (*states*) and occasional *rewards* (or punishment)
- Generation and evaluation of a *policy* i.e. a mapping from states to *actions*
- A form of active learning
- Neither really supervised nor unsupervised
- A microcosm for the entire AI problem

"The use of punishments and rewards can at best be a part of the teaching process" (A. Turing)

Russell and Norvig: AI, Ch.21

What means "RL" (in Machine Learning)?

- Learning given only percepts (*states*) and occasional *rewards* (or punishment)
- Generation and evaluation of a *policy* i.e. a mapping from states to *actions*
- A form of active learning
- Neither really supervised nor unsupervised
- A microcosm for the entire AI problem

"The use of punishments and rewards can at best be a part of the teaching process" (A. Turing)

Russell and Norvig: AI, Ch.21

What else is RL?

Arthur Samuel (1959): Computer Checkers

- **Search tree:** board positions reachable from the current state. Follow paths as indicated by a
- **Scoring function:** based on the position of the board at any given time; tries to measure the chance of winning for each side at the given position.
- Program chooses its move based on a minimax strategy
- **Self-improvement:** Remembering every position it had already seen, along with the terminal value of the reward function. It played thousands of games against itself as another way of learning.
- First to play any board game at a relatively high of level
- The earliest successful machine learning research

wikipedia and Russell and Norvig: AI, Ch.21

More history

- SNARC: Stochastic Neural Analog Reinforcement Calculator (M. Minsky, 1951)
- A. Samuel (1959) Computer Checkers
- B. Widrow and T. Hoff (1960) adapted the D. O. Hebb's neural learning rule (1949) for RL: delta rule
- Cart-pole problem (D. Michie and R.A. Chambers, 1968)
- Relation between RL and MDP (P. Werbos, 1977)
- A. Barto, R. Sutton, P. Brouwer (1981) Associative RL
- $\mathcal{Q}$-learning (C.J.C.H. Watkins, 1989)
-

Russell and Norvig: AI, Ch.21

Aspects of RL (outlook)

- MAB, MDP, DP, MC, SARSA, TD(λ), SMDP, POMDP, ...
- Exploration
- Active learning and machine learning
- Structure of state and action spaces
- Continuous domains: Function approximation
- Complexity, optimality, efficiency, numerics
- [psychology, neuroscience]

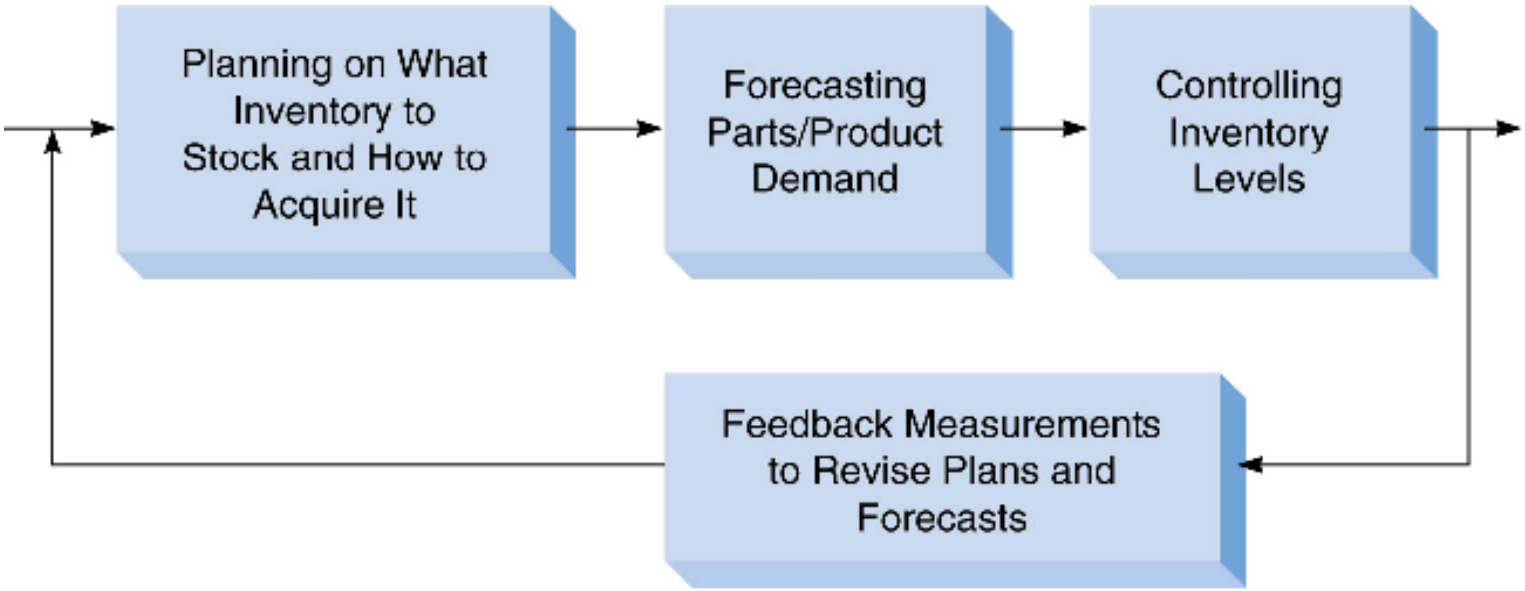
Generic Examples

- Motor learning in young animals: No teacher. Sensorimotor connection to environment.
- Language acquisition
- Learning to
  - drive a car
  - hold a conversation
  - learning to cook
  - play games: backgammon, checkers, chess
  - play a musical instrument
- Applications
  - problem solving: Find a policy (states $\rightarrow$ actions) that maximises reward
  - scheduling, control, operations research, HCI, economics

Practical approach to the problem

- Many ways to approach these problems
- Unifying perspective: *Stochastic optimisation over time*
- Given
  - Environment to interact with
  - Goal
- Formulate cost (or *reward*)
- *Objective*: Maximise rewards over time
- The catch: Reward signal not always available, but optimisation is over time (selecting entire paths)

Example: Inventory Control



- Objective: Minimise total inventory **cost**
- Decisions:
  - How much to order?
  - When to order?

Components of Total Cost

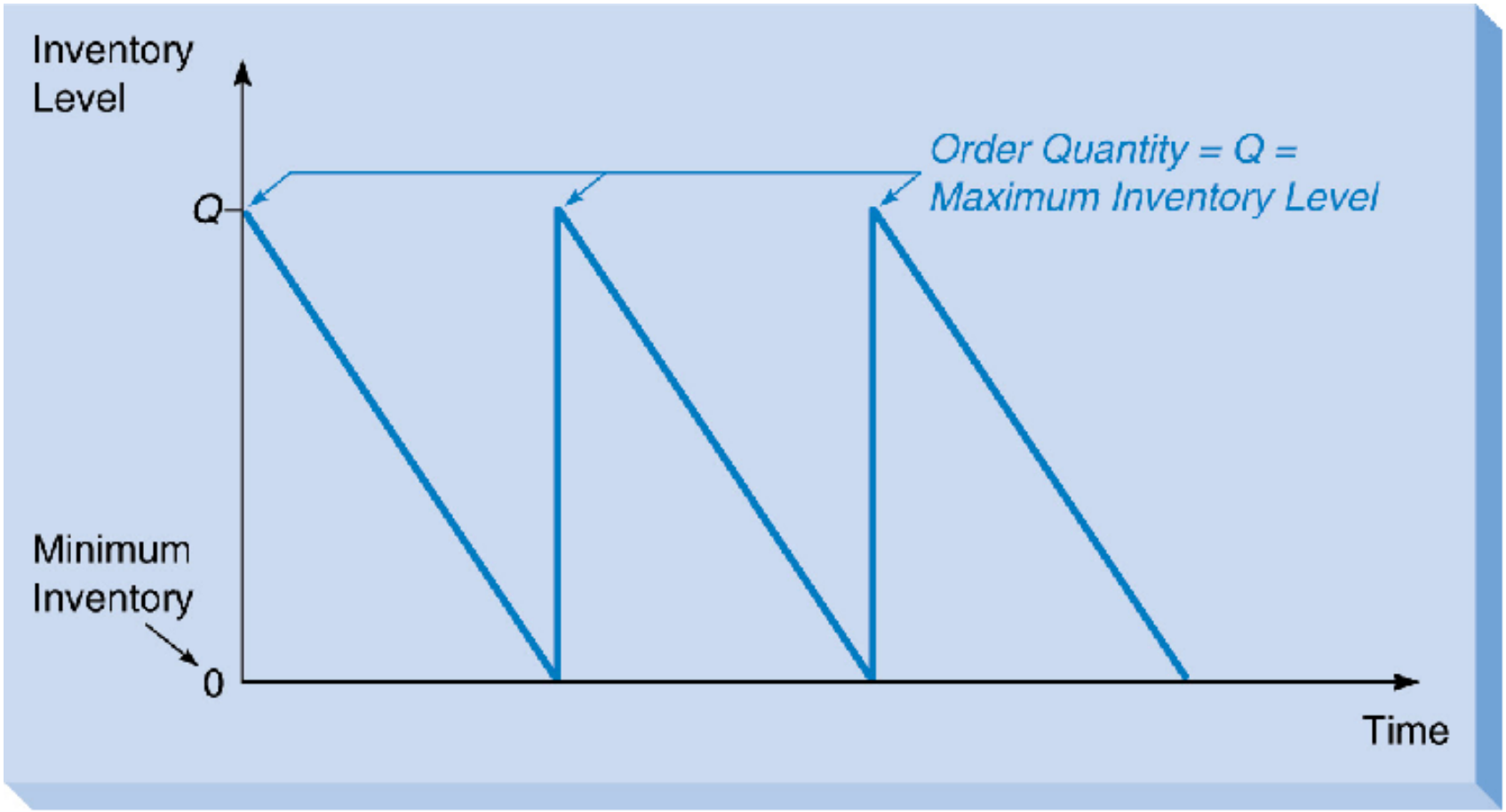
1. Cost of items
2. Cost of ordering
3. Cost of carrying or holding inventory
4. Cost of stockouts
5. Cost of safety stock (extra inventory held to help avoid stockouts)

The Economic Order Quantity Model - How Much to Order?

Assumptions

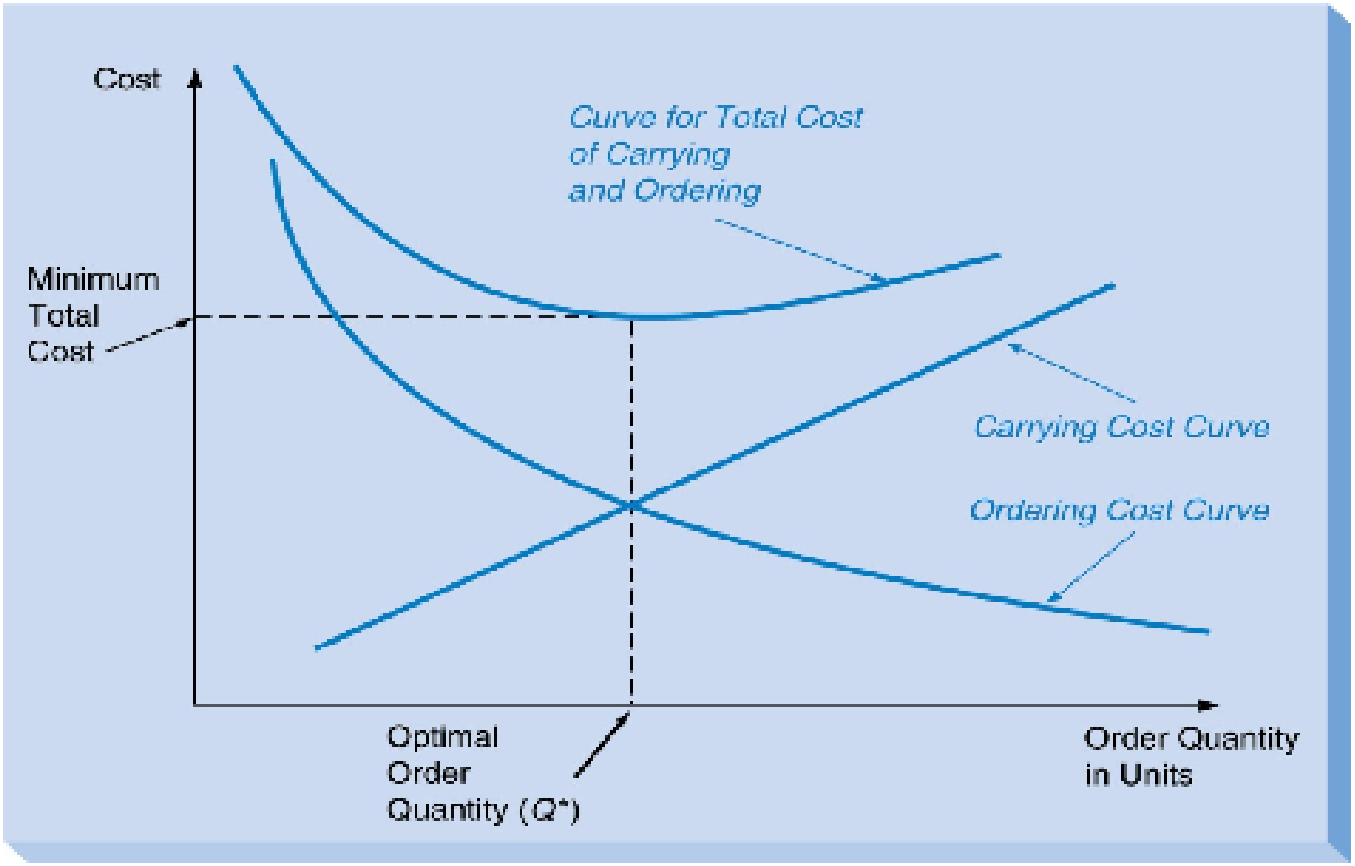
1. Demand is known and constant
2. Lead time is known and constant
3. Receipt of inventory is instantaneous
4. Quantity discounts are not available
5. Variable costs are limited to: ordering cost and carrying (or holding) cost
6. If orders are placed at the right time, stockouts can be avoided

Inventory Level Over Time Based on EOQ Assumptions



Economic order quantity, Ford W. Harris, 1913

EOQ Model Total Cost



At optimal order quantity (Q*): (Carrying cost)' = (Ordering cost)'

$$Q^* = \sqrt{\frac{2DC_o}{C_h}}$$

D: demand, C_o, C_h: costs

Realistically, how much to order

If these assumptions didn't hold?

- Demand is ~~known~~ and ~~constant~~
- Lead time (latency) is ~~known~~ and ~~constant~~
- Receipt of inventory is ~~instantaneous~~
- Quantity discounts are ~~not~~ available
- ~~Variable costs are limited to~~: ordering cost and carrying (or holding) cost
- If orders are placed at right time, stockouts ~~can be avoided~~

The result may require a more detailed stochastic optimisation.

Properties of RL learning tasks

- Does not assume that you know the model of the environment
- Associativity: Value of an action depends on state
- Active learning: Environment's response affects our subsequent actions and thus future responses
- Delayed reward: We find out the effects of our actions later
- Credit assignment problem: Upon receiving rewards, which actions were responsible for the rewards?

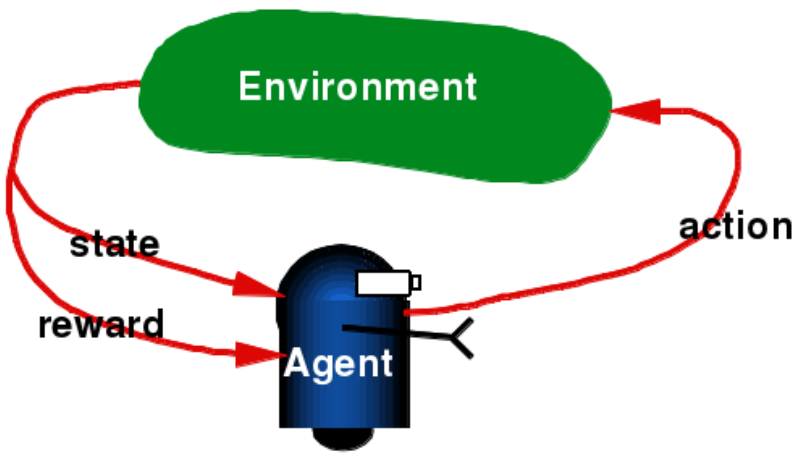
Conclusions

- Stochastic Optimisation – make decisions! Over time; may not be immediately obvious how we're doing
- Some notion of cost/reward is implicit in problem – defining this, and constraints to defining this, are key!
- Often, we may need to work with models that can only generate sample traces from experiments

Summary: The Setup for RL

Agent is:

- Temporally situated
- Continual learning and planning
- Objective is to affect the environment by state-dependent actions
- Environment is uncertain, stochastic



Key Features of RL

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

Relation to Metaheuristic Optimisation (MO)

- "Natural computing" including algorithms such as GA, ES, DE, PSO, or ACO
- For example, genetic algorithms (GA) maximise a fitness function by selecting, mutating and reproducing certain individuals in a population
- The individuals are described as strings over an alphabet, e.g. {G,A,T,C}
- The gradient of the fitness function is not known
- The population is a sample of points in the space of all strings

Difference between RL and Metaheuristic Optimisation

	RL	MO
representation	usually a single agent that must be restarted	population
learning steps	rewards is received after each action	fitness is evaluated in each generation
dynamics	Markovian state transitions	population moves in configuration space
global search	trial and error guided by earlier experiences	trial and error, may be guided by correlations
local search	policy gradient	hill climbing

Both are stochastic optimisation algorithms. Both can find their resp. global optimum under certain (though artificial) conditions.

RL provides a theoretical framework for Markovian worlds; MO is more general but its function is less understandable.

Many combinations are possible, e.g. collective RL or MO of the hardware or architecture of a robot that learns by RL

Relation to Dynamic Programming (DP)

- Dynamic programming is a method for solving a complex problem by breaking it down into a collection of simpler subproblems (wikipedia).
- Often this means: Separating the problem into one part for the next time step and one part for all remaining time steps.
- In RL, we are interested in problem where a good model is unavailable
- An RL agent cannot compare different solutions unless it acts such as to actually generate these solutions.
- When we design or analyse RL algorithms, we will often refer to the DP framework.

What is Reinforcement Learning?

- A paradigm in Artificial Intelligence
- Goal-oriented learning (Maximise reward!)
- Can be thought of as a stochastic optimisation over time
- A practical algorithm for dynamic programming problems
- Learning about, from, and while interacting with an external environment
- Learning what to do — how to map situations to actions — so as to maximise a numerical reward signal

A bit of Maths

Geometric series

$$s_n = \sum_{i=0}^{n} \alpha^i$$

Sum of an infinite geometric series

$$\lim_{n\to\infty} s_n = \frac{1}{1-\alpha}$$

Averages over a time series $\{x_n\}$: math. expectation

Moving average (practically)

$$\bar{x}_n = \frac{1}{k} \sum_{i=n-k+1}^{n} x_i$$

Weighted (moving) average

$$\bar{x}_n = \frac{1}{\sum_{j=n-k+1}^{n} \alpha_{n-j}} \sum_{i=n-k+1}^{n} \alpha_{n-i} x_i = \frac{1}{\sum_{j=0}^{k-1} \alpha_j} \sum_{i=0}^{k-1} \alpha_i x_{n-i}$$

Exponentially weighted average $\bar{x}_n = (1-\alpha) \sum_{i=0}^{\infty} \alpha^i x_{n-i}$