BLG456E
Robotics
Intro to Reactive Robot Learning

<u>Lecture Contents:</u>
- Why learning.
- Supervised learning.
- Basic reactive controller learning.
- Lookup-table learning.
- Introducing time & credit assignment.

Lecturer:	Damien Jade Duff
Email:	djduff@itu.edu.tr
Office:	EEBF 2316
Schedule:	http://djduff.net/my-schedule
Coordination:	http://ninova.itu.edu.tr/Ders/4709

Some slides taken/adapted from Dr. Sanem Sarıel-Talay

What is learning?

- 1. **Learning**: getting better at doing things from experience.

- 2. **Learning**: When a program C improves its performance in T according to P after incorporating E.
  - Computer program C.
  - Class of tasks T.
  - Performance measure P.
  - Experience E.

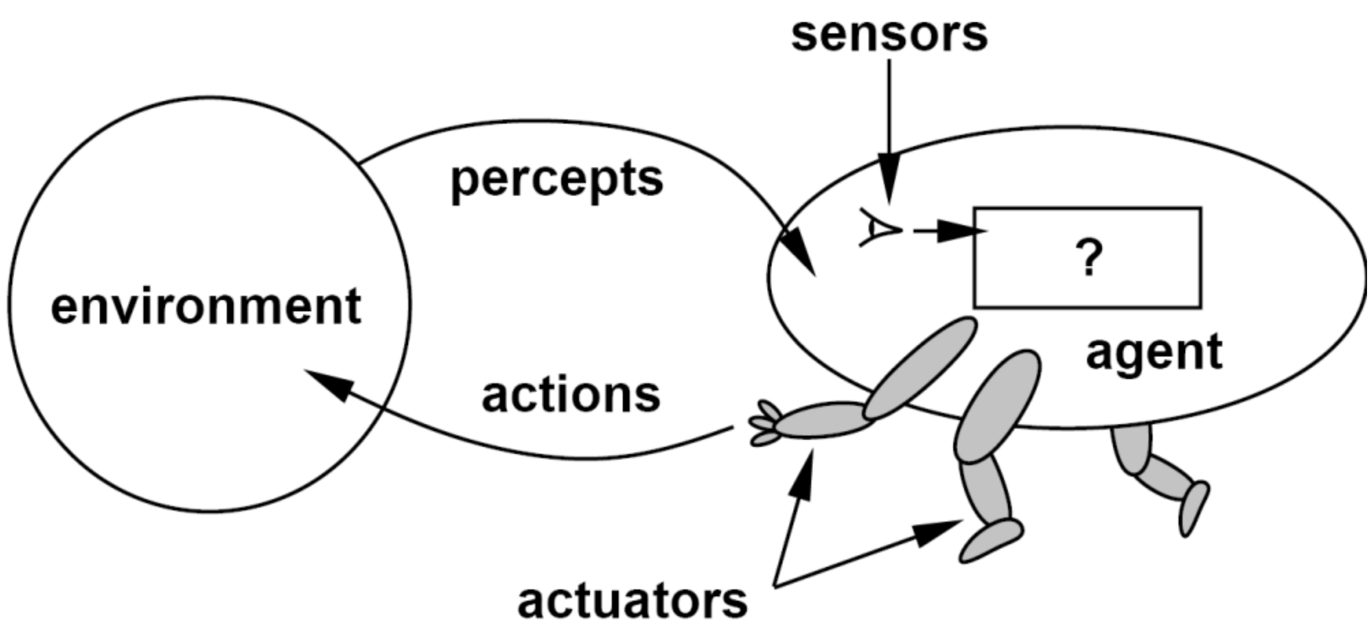
Why learning?

- Not fully specified problems.
  - e.g. unknown environments.
  - e.g. incomplete models.
- Vast amounts of data.
- *Why calibration*?
- Leverage "knowledge" of world itself.
- Adapt.

Successful Examples

- Learning to drive an autonomous vehicle.
- Learning to walk.
- Learning to classify objects.
- Learning to play world-class backgammon.
- Learning to do X better.

Recall the sense-action loop



Let's learn $f : P \rightarrow A$

A percept-action mapping

BLG456E
Robotics
Intro to Reactive Robot Learning

Lecture Contents:
- Why learning.
- Supervised learning.
- Basic reactive controller learning.
- Lookup-table learning.
- Introducing time & credit assignment.

Lecturer:	Damien Jade Duff
Email:	djduff@itu.edu.tr
Office:	EEBF 2316
Schedule:	http://djduff.net/my-schedule
Coordination:	http://ninova.itu.edu.tr/Ders/4709

Some slides taken/adapted from Dr. Sanem Sarıel-Talay

Kinds of learning

- “*Supervised*”
  - Output (Y) provided for input (X). Learn relationship.
- “*Unsupervised*”
  - Full input/output examples not provided (just X). Learn structure.
- “*Reinforcement learning*”
  - Only occasional feedback.

Supervised learning

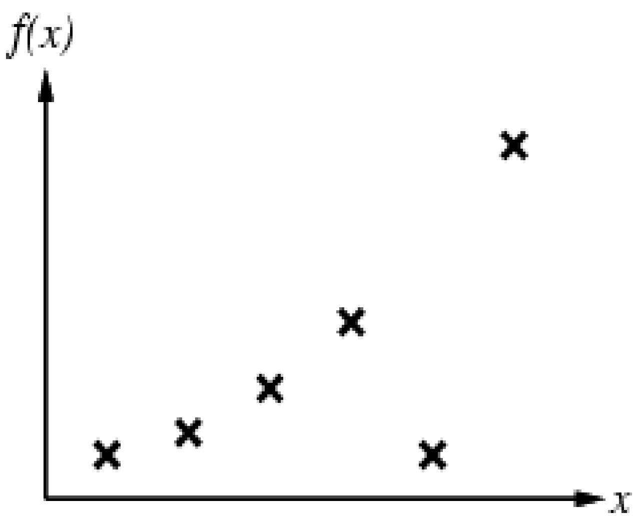
- Given input/output pairs (X,Y).
- Learn a mapping $f:X \rightarrow Y$.
- e.g. Sense-action learning from demonstration.

iCub learning to catch:http://www.youtube.com/watch?v=EZDObbQYFFM

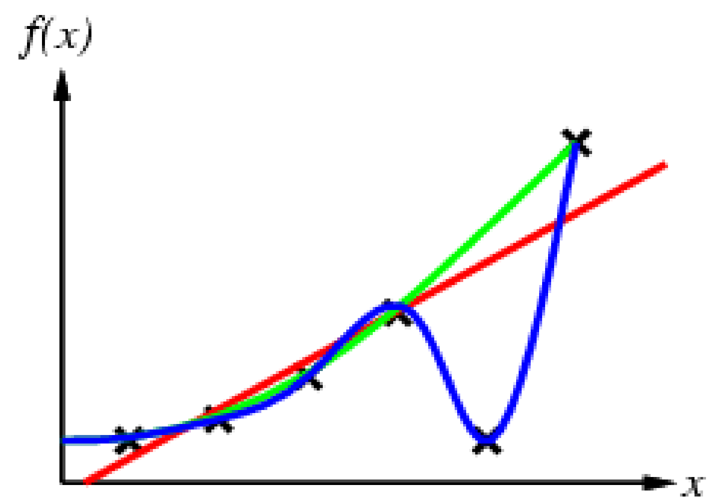
Supervised learning

- Construct hypothesis to agree with training set.
  - Consistent.
- Prefer simplest consistent hypothesis.
  - Ockham’s razor.

X , Y examples



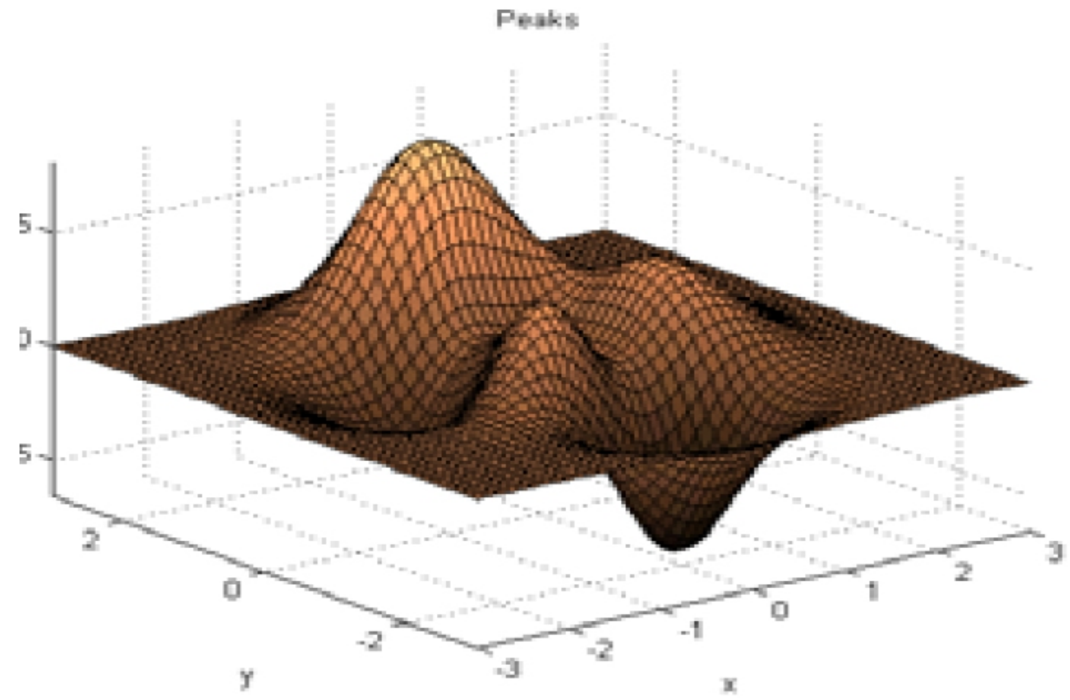
Some possible learnt functions
(hypotheses)



(curve fitting example)

Learning is searching

- All possible solutions exist in the solution space.
- Finding a good or optimal solution.



Each hypothesis curve can be a point in the search space

Representing sensory-action mapping

$$f : \boldsymbol{p} \rightarrow \boldsymbol{a}$$

f is a function from percepts to actions.

Percepts and actions
can be sets of numbers - vectors

$$\text{e.g.} \quad \boldsymbol{p} = \begin{bmatrix} 5.0 \\ -1.3 \\ 3.0 \\ 1.2 \end{bmatrix} \quad \boldsymbol{a} = \begin{bmatrix} 45 \\ 0.3 \end{bmatrix}$$

4 laser ranges, two motors

Representing sensory-action mapping

e.g. $\boldsymbol{p}=\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \quad \boldsymbol{a}=\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}=\boldsymbol{f}(p_1, p_2)$$

f must be parameterisable.

e.g.

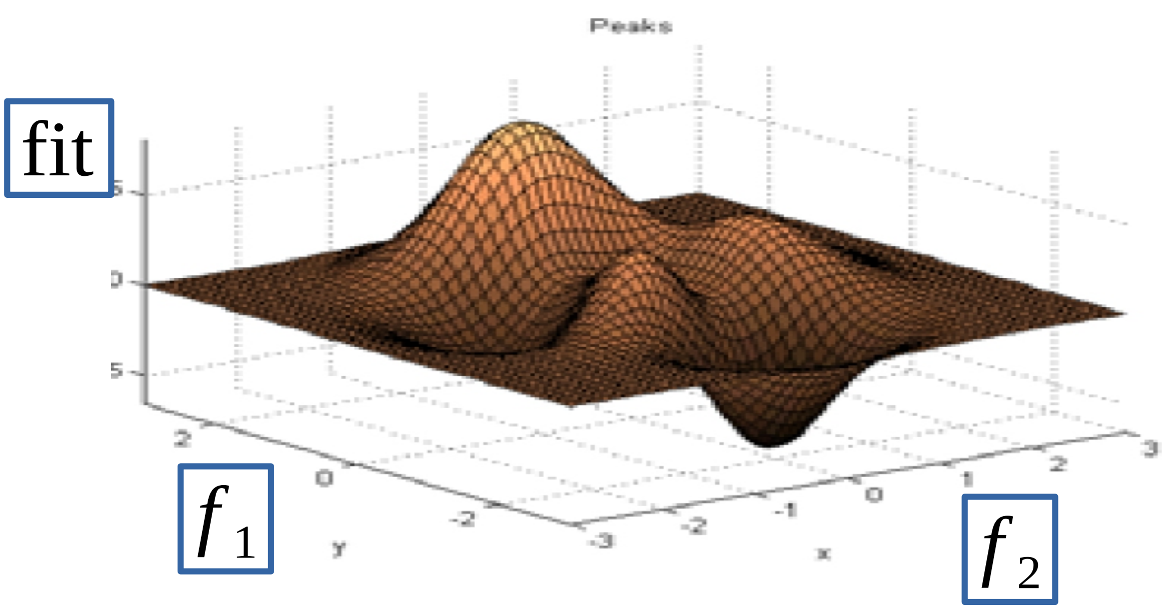
$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}=\boldsymbol{f}(p_1, p_2)=\begin{bmatrix} f_1 p_1 p_2 + f_2 p_1^2 + f_3 p_2^2 + f_4 p_1 + f_5 p_2 + f_6 \\ f_7 p_1 p_2 + f_8 p_1^2 + f_9 p_2^2 + f_{10} p_1 + f_{11} p_2 + f_{12} \end{bmatrix}$$

(Polynomial function – but learning this would be harder)

Learning is searching for parameters of function

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \boldsymbol{f}(p_1, p_2) = \begin{bmatrix} f_1 p_1 p_2 + f_2 p_1^2 + f_3 p_2^2 + f_4 p_1 + f_5 p_2 + f_6 \\ f_7 p_1 p_2 + f_8 p_1^2 + f_9 p_2^2 + f_{10} p_1 + f_{11} p_2 + f_{12} \end{bmatrix}$$

Task: Find the $f_1, f_2, \ldots, f_{11}, f_{12}$ that best fit data.



Learning by reducing error (improving fit)

For a certain percept:

$$\boldsymbol{p} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$$

Our learner has a current prediction:

$$\hat{\boldsymbol{a}} = \begin{bmatrix} \widehat{a}_1 \\ \widehat{a}_2 \end{bmatrix} = \boldsymbol{f}(p_1, p_2)$$

And an observed action:

$$\boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

Error:

$$\epsilon = \|\boldsymbol{a} - \hat{\boldsymbol{a}}\|^2 = \left\| \begin{bmatrix} a_1 - \widehat{a}_1 \\ a_2 - \widehat{a}_2 \end{bmatrix} \right\|^2$$

Simple sensory-action mapping: lookup table

$a_1 = f_1(p_1,p_2)$		p_2			
		0.0-1.0	1.0-2.0	2.0-3.0	3.0-4.0
p_1	0.0-1.0	3.4	2.9	2.4	2.4
	1.0-2.0	4.6	4.8	4.9	4.8
	2.0-3.0	4.4	5.3	5.3	4.4
	3.0-4.0	1.0	0.8	0.9	1.0

$f_1(0.7,1.3)=2.9$ $f_2(0.7,1.3)=140$

$a_2 = f_2(p_1,p_2)$		p_2			
		0.0-1.0	1.0-2.0	2.0-3.0	3.0-4.0
p_1	0.0-1.0	86	140	203	270
	1.0-2.0	95	145	207	265
	2.0-3.0	105	156	223	254
	3.0-4.0	115	160	231	233

Other possible representations of the function f?

- Overlapping bins (CMAC).
- Neural network.
- Exemplar-based (nearest-neighbour search).
- Kernel function.

…

Reduce error by changing the function parameters (learning)

$a_1 = f_1(p_1,p_2)$		p_2			
		0.0-1.0	1.0-2.0	2.0-3.0	3.0-4.0
p_1	0.0-1.0	3.4	2.9	2.4	2.4
	1.0-2.0	4.6	4.8	4.9	4.8
	2.0-3.0	4.4	5.3	5.3	4.4
	3.0-4.0	1.0	0.8	0.9	1.0

If $p_1 = 0.4$ and $p_2 = 3.4$ and action $a_1 = 3.2$ is <u>given.</u>

$$\boldsymbol{a_1 = 3.2} \qquad \boldsymbol{\hat{a}_1 = 2.4}$$

$$\epsilon = 0.8$$

$(\hat{a}_1 = f_1(0.4, 3.4) = 2.4$ and the error is $\epsilon = 3.2 - 2.4 = 0.8)$

Reduce error by changing the function parameters (learning)

$a_1 = f_1(p_1,p_2)$		p_2			
		0.0-1.0	1.0-2.0	2.0-3.0	3.0-4.0
p_1	0.0-1.0	3.4	2.9	2.4	2.4
	1.0-2.0	4.6	4.8	4.9	4.8
	2.0-3.0	4.4	5.3	5.3	4.4
	3.0-4.0	1.0	0.8	0.9	1.0

2.44

If $p_1 = 0.4$ and $p_2 = 3.4$ and action $\boldsymbol{a_1 = 3.2}$ is observed.

$(\hat{a}_1 = f_1(0.4, 3.4) = 2.4$ and the error is $\epsilon = 3.2 - 2.4 = 0.8)$

How to reduce this error by changing f ?

Move $f_1(0.4, 3.4)$ towards 3.2

$$f_1[0.0-1.0, 3.0-4.0] \leftarrow f_1[0.0-1.0, 3.0-4.0] + 0.05 \cdot 0.8$$

Let's convert that to code.

We need:

- A lookup table (array) to represent f.
  - Dimensions = number of parameters.
  - Size in each dimension = number of ranges (bins).
- To observe demonstrated percept-action pairs.
  - e.g. ROS subscription to topics.
- To learn by updating the table according to these.
  - Simple array lookup and increment.
- **To use the table by looking up action from perception.**
  - Simple array lookup.

Problems with supervised learning

- Demonstrator not always available.
  - e.g. learning to run away from lions.
- Demonstration not always applicable.
  - e.g. human demonstrator with human shape.
- “Labelled data” not always available.
  - e.g. recognising all the objects in the world.

BLG456E
Robotics
Intro to Reactive Robot Learning

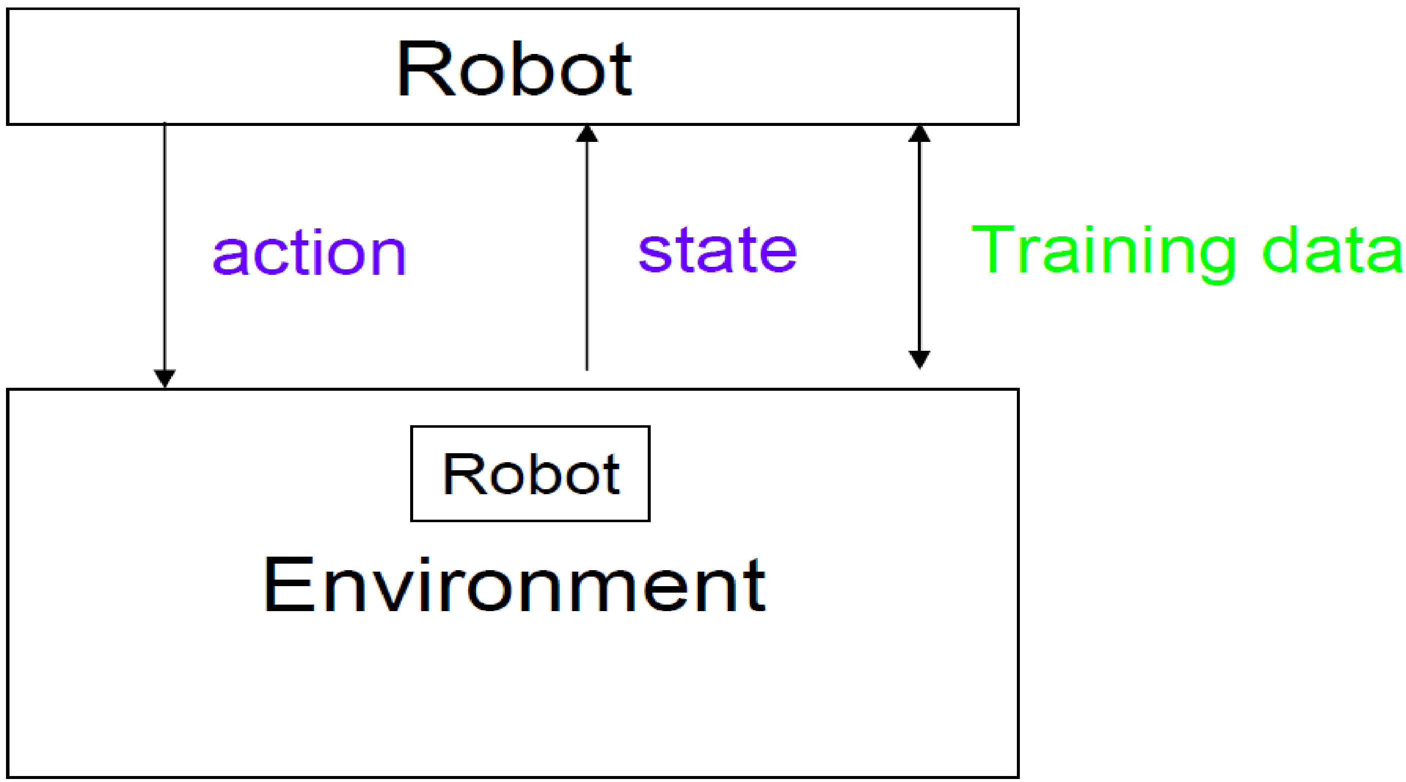
Lecture Contents:

- Why learning.
- Supervised learning.
- Basic reactive controller learning.
- Lookup-table learning.
- Introducing time & credit assignment.

Lecturer:	Damien Jade Duff
Email:	djduff@itu.edu.tr
Office:	EEBF 2316
Schedule:	http://djduff.net/my-schedule
Coordination:	http://ninova.itu.edu.tr/Ders/4709

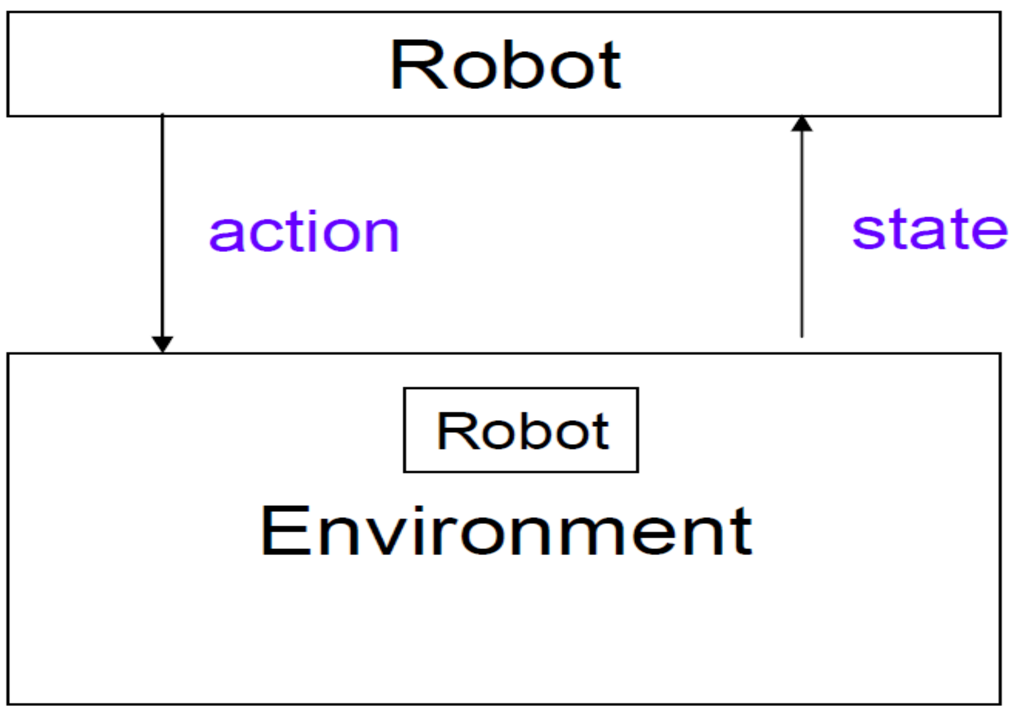
Supervised learning

- Pairs of state-action (s, a) given as training data.
- robot learns appropriate action for a state.



Unsupervised learning

- The robot gets input data x_1, x_2, … x_n
- Construct a representation of x for reasoning, decision making, classification, …



e.g. clustering

Temporal difference learning

- Only occasional feedback.