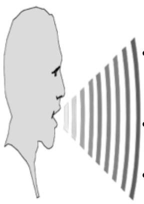
EEL5840: Elements of Machine Intelligence

Robobug

Announcements



- Announcements:
  - > Midterm Date
    - Thu. Oct. 8th
- Today's Handouts in WWW:
  - > Outline Class 16
- Web Site
  - > www.mil.ufl.edu/eel5840
  - > Software and Notes

EEL5840: Elements of Machine Intelligence

Robobug

ANN Websites

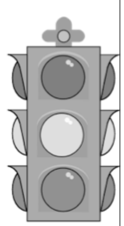
- http://aima.cs.berkeley.edu/
- http://www.ibm.com/developerworks/library/l-neural/
- http://www.dia.fi.upm.es/~jamartin/download.htm
- http://www.answermysearches.com/four-free-neural-network-libraries-for-python/195/

EEL5840: Elements of Machine Intelligence

Robobug

Today's Menu

- State Machines
  - > Representing the Environment by Feature Vectors
  - > Elman Networks
  - > Iconic Representations
  - > Blackboard Systems (maybe)



EEL5840: Elements of Machine Intelligence

Robobug

State Machines

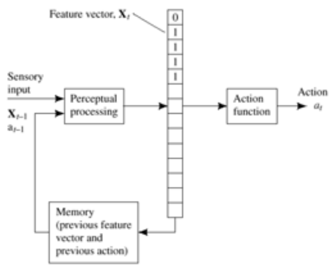
- Representing the Environment by Feature Vectors
  - > A robot perceives the environment through its sensory array, therefore, the sensory feature vector can be thought of as a representation of the environment.
  - > The robot reacts (performs an action) based on this feature vector.
  - > Sensory limitations of the agent preclude completely accurate representation of the state of the world via feature vectors—however the accuracy can be improved by taking into account previous sensory history.
  - > State Machines: machines where the representation of environmental state at step *t+1* is a function of the sensory input at *t+1*, and the representation of the previous state and action at time *t*. By definition state machines must have memory in which to store a model of the environment, and there must be an initial representation at time *t=0* (e.g., preloaded by the designer).

Robobug

EEL5840: Elements of Machine Intelligence

State Machines

- Representing the Environment by Feature Vectors
  > A suitable model is as follows:



The key point is that x_{t+1} depends also on x_t , the feature vector from the previous step

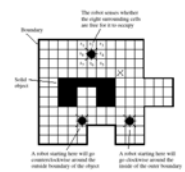
Figure 5.1 A State Machine Representation

Robobug

EEL5840: Elements of Machine Intelligence

State Machines

- Representing the Environment by Feature Vectors
  > The environment is always arbitrarily complex which results in an imperfect, at best, representation. The task of the agent is one of arriving at an adequate representation insofar as the agent's tasks are concerned.
  > State Machine Example: The boundary-following robot



The sensory input vector *s* in this example is given by the eight sensory inputs {s_1, s_2, s_3,…,s_8}.

Figure 2.1

Robobug

EEL5840: Elements of Machine Intelligence

State Machines

- Representing the Environment by Feature Vectors
  > Suppose the robot cannot sense the four diagonally adjacent cells (s_1, s_3, s_5, s_7). Recall s_i=1 if the cell is not free. Is it possible to still do wall-following with this impairment? Yes, assuming we can use a state machine representation inside of the robot.
  > Let us define a feature vector $w_i = s_i$ for i=2,4,6,8. In place of the four sensory inputs that are absent we will substitute four features, w_1, w_3,w_5, w_7.
  $w_1(t)=1$ iff $w_2(t-1)=1$ and *action*(t-1)=*east*, else *0*.
  $w_3(t)=1$ iff $w_4(t-1)=1$ and *action*(t-1)=*south*, else *0*.
  $w_5(t)=1$ iff $w_6(t-1)=1$ and *action*(t-1)=*west*, else *0*.
  $w_7(t)=1$ iff $w_8(t-1)=1$ and *action*(t-1)=*north*, else *0*.

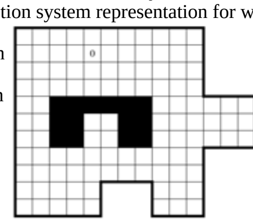
Robobug

EEL5840: Elements of Machine Intelligence

State Machines

- Representing the Environment by Feature Vectors
  A possible production system representation for wall-following is:

$w_2/w_4 \rightarrow$ east
$w_4/w_6 \rightarrow$ south
$w_6/w_8 \rightarrow$ west
$w_8/w_2 \rightarrow$ north
$w_1 \rightarrow$ north
$w_3 \rightarrow$ east
$w_5 \rightarrow$ south
$w_7 \rightarrow$ west
1 $\rightarrow$ north



At point *t=1*, $s_i=0$ for i=2,4,6,8. Thus, $w_i=0$ for i=2,4,6,8. $w_1=0$ *at t=1* since $w_2(0)=0$; and w_3, w_5 and w_7 are all *0*. The robot moves north.

$w_i = s_i$ for i=2,4,6,8.
$w_1(t)=1$ iff $w_2(t-1)=1$ and *action*(t-1)=*east*, else *0*.
$w_3(t)=1$ iff $w_4(t-1)=1$ and *action*(t-1)=*south*, else *0*.
$w_5(t)=1$ iff $w_6(t-1)=1$ and *action*(t-1)=*west*, else *0*.
$w_7(t)=1$ iff $w_8(t-1)=1$ and *action*(t-1)=*north*, else *0*.

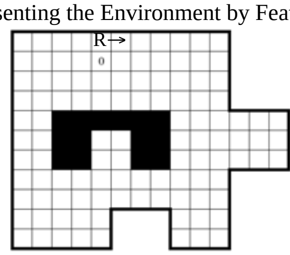
Robobug

EEL5840: Elements of Machine Intelligence

State Machines

- Representing the Environment by Feature Vectors

$w_2/w_4 \rightarrow$ east
$w_4/w_6 \rightarrow$ south
$w_6/w_8 \rightarrow$ west
$w_8/w_2 \rightarrow$ north
$w_1 \rightarrow$ north
$w_3 \rightarrow$ east
$w_5 \rightarrow$ south
$w_7 \rightarrow$ west
$1 \rightarrow$ north



At point $t=2$,
$s_2=1$ and $s_i=0$ for i=4,6,8.
$w_2=1$ and $w_i=0$ for i=4,6,8
$w_1(2)=0$ since $w_2(1)=0$
$w_3(2)=0$ since $w_4(1)=0$
$w_5(2)=0$ since $w_6(1)=0$
$w_7(2)=0$ since $w_8(1)=0$
$w_2/w_4=1$ ∴ east
The robot moves east.

$w_i = s_i$ for i=2,4,6,8.
$w_1(t)=1$ iff $w_2(t-1)=1$ and $action(t-1)=east$, else 0.
$w_3(t)=1$ iff $w_4(t-1)=1$ and $action(t-1)=south$, else 0.
$w_5(t)=1$ iff $w_6(t-1)=1$ and $action(t-1)=west$, else 0.
$w_7(t)=1$ iff $w_8(t-1)=1$ and $action(t-1)=north$, else 0.

Robobug

EEL5840: Elements of Machine Intelligence

State Machines

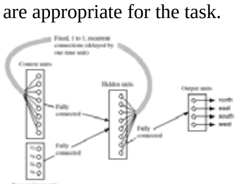
- Representing the Environment by Feature Vectors
  - > The wall-following robot of Chapter 2 achieved the same behavior without memory—that is without previous inputs, features or actions. If all the important aspects of the environment can be sensed at the time the agent needs to know them, there is no reason to retain a model of the environment in memory.
  - > Sensory abilities are always limited in some way, and thus agents equipped with memory (state machine agents) that can can store models of the environment will usually be able to perform tasks that SR agents cannot.
  - > Note the tradeoff between memory and non-memory is similar to the tradeoffs between combinatorial and sequential digital logic circuits.

Robobug

EEL5840: Elements of Machine Intelligence

State Machines

- Elman Networks
  - > A special type of recurrent neural network [Elman 1990] used to learn to compute a feature vector and an action from a previous feature vector and sensory inputs.
  - > Nilsson illustrates how this might be done for the wall-following robot that senses only four cells to the north, east, south and west. The net learns to store those properties of previously sensed data that are appropriate for the task.



This network has eight hidden units: one for each of the w_i features. The input vector x consists of the four sensory inputs (s_2,s_4,s_6,s_8) plus the values of the eight hidden units from one step earlier ($t-1$). These extra inputs are called *context units*, and they allow the net to learn on past data, i.e., the robot's context.

Robobug

EEL5840: Elements of Machine Intelligence

State Machines

- Elman Networks
  - > The net has four outputs, one for each possible action and each action is computed from the same set of hidden units (corresponding to the features).
  - > Training data is such that only one output has a desired output of 1; the others have desired output of 0.
  - > The training data is obtained by running an expert wall-following teacher program and presenting the same sequence of sensory inputs and actions to the network. (This corresponds to a sequence of sensory inputs labeled by the appropriate corresponding output action).
  - > After training we select the output unit with the largest output as the robot action.

Robobug

EEL5840: Elements of Machine Intelligence

State Machines

- Elman Networks
  > An Elman network is a special case of a more general artificial neural network (ANN) and therefore all the techniques for ANNs can be used to train and design Elman Nets.

  > In particular, training can be accomplished with backpropagation (quickprop, steepest descent, or conjugate gradient methods) because the backward directed weights from the hidden units to the context units are fixed and the context units can be treated as just another set of inputs.

Robobug

EEL5840: Elements of Machine Intelligence

State Machines

- Iconic Representations
  > The representation of the environment or world by data structures such as maps, queues, lists, etc. These can all be thought of as simulations of important aspects of the environment, e.g., *analogical representations.*

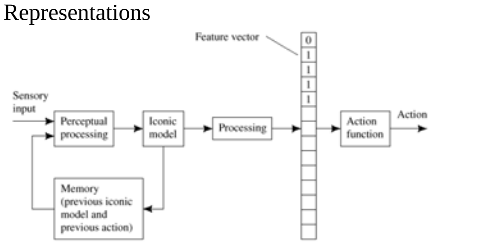
  > When an agent uses an iconic representation it must still compute actions appropriate to its task and to the present state of the environment as represented in the model. Reactive agents react to the model in the same way that agents without memory react to sensory stimuli: they compute features of the data structure.

Robobug

EEL5840: Elements of Machine Intelligence

State Machines

- Iconic Representations



The sensory information is first used to update the iconic model. Processing then extracts features that are needed by the action subsystem. Actions may include those that change the iconic model as well as those that affect the actual environment. The features must be adequate representations of the environment with respect to the action set of the agent.

Robobug

EEL5840: Elements of Machine Intelligence

State Machines

- Iconic Representations
  > Suppose we use a matrix of free and non-free cells to represent the grid-space robot. The effect is as if the robot had sensors that could sense whether or not any of the cells in the grid space were free.

	1	1	1	1	1	1	1	?
1	0	0	0	0	0	0	0	?
1	0	0	0	0	0	0	0	?
1	0	0	0	0	0	0	0	?
1	0	0	0	0	0	0	0	?
1	0	0	*R*	0	0	0	0	?
1	0	0	0	0	0	0	0	?
1	0	0	0	0	0	0	0	?
1	0	0	0	0	0	0	0	?
1	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?

Free cells are represented by 0's in the corresponding array element; non-free cells are represented by 1's; unknown cells are represented by ?'s; and the robot's own location by an *R*. If the robot is designed to go to the closest wall and then do wall following, then this data structure should evoke the action →*west*. Once the action west is performed, the agent must update the data structure to change its own position and to take into account new sensory data.

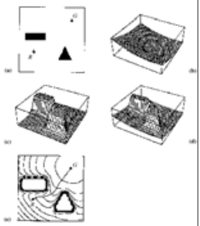
EEL5840: Elements of Machine Intelligence

Robobug

State Machines

- Iconic Representations

  Another type of action computation mechanism uses a model in the form of an artificial potential field [Latombe 1991]. This is used extensively in controlling robotic manipulators. The robot's environment is represented as a two-dimensional potential field, with higher-dimensional fields used when there are many degrees of freedom to be controlled. The potential field is the sum of an *attractor* and a *repulsive* field. The attractive field is associated with the goal location and obstacles are associated with a repulsive field.



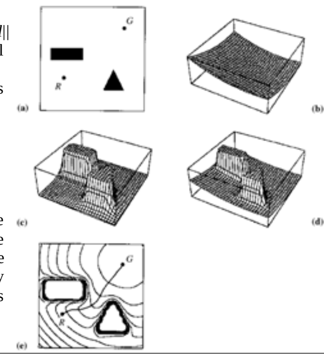
EEL5840: Elements of Machine Intelligence

Robobug

State Machines

A typical attractor is given by:
$p_a(\boldsymbol{x}) = k_1 d(\boldsymbol{x})^2$ and $d(\boldsymbol{x})=||\boldsymbol{x}\text{-}goal||$ and it has a minimum at the goal location.
A typical repulsive function is given by:
$p_r(\boldsymbol{x}) = k_2/d_0(\boldsymbol{x})^2$ and
$d_0(\boldsymbol{x})=||\boldsymbol{x}\text{-}obstacle||$
The total potential is given by:
$p = p_a + p_r$

Motion is directed along the gradient of the potential field. The potential field function *p* can be precomputed and stored in memory or calculated incrementally as needed.



EEL5840: Elements of Machine Intelligence

Robobug

The End!