

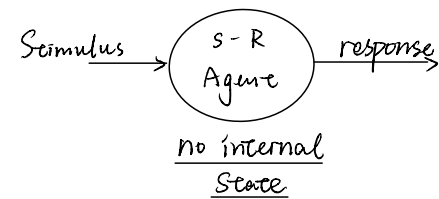
Agent

"Perform certain task in some environment autonomously"

An agent has

1. Perception
2. Actions
3. Goal

Stimulus-Response Agent

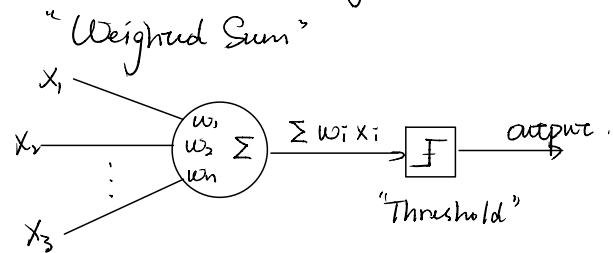


Action Function

sensors $\xrightarrow{\text{mapping}}$ action

Learning action function with TLUs

TLU: Threshold Logic Unit



"Linear separable function" can be implemented as TLU

Error-Correction Procedure

$$W \leftarrow W + c(d - f) \times$$

$\uparrow$
 "label" (TLU output)
 learning rate

Neurons / Activation function / Artificial Neuron Network

Same in COMP 6211

Designing Action Function with "Production System"

Production System

" $c \rightarrow a$ "

where c is a "condition", a is an "action"

production system is a set of " $c \rightarrow a$ ".

Learning Action Function With Genetic Programming

GP: Genetic Programming

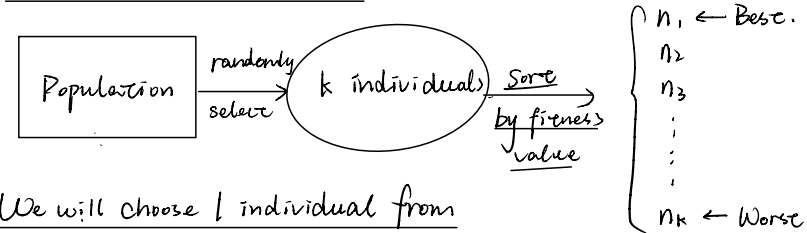
general ideas:

1. "What is legal program?"
2. Define a fitness function
3. Select a set of "legal program"s as generation 0.
4. produce next generation until a desired program is generated

How to generate next generation?

- ① Copy. copy the elites straightly to next generation.
- ② Crossover. mix up two parent's genes.
- ③ Mutation. mutate certain genes randomly

"Tournament Selection"



We will choose 1 individual from the chosen k individuals

Choose the "best" directly? No. Since we want variety in generations.

$\Rightarrow$ We choose n_1 with probability p , n_2 with $p(1-p)$...

$$\Rightarrow \begin{cases} n_1 \rightarrow p \\ n_2 \rightarrow p(1-p) \\ n_3 \rightarrow p(1-p)^2 \\ \vdots \end{cases}$$

Generation 0: 5000 random programs.

Procedure for producing generation $(n+1)$ from generation n :

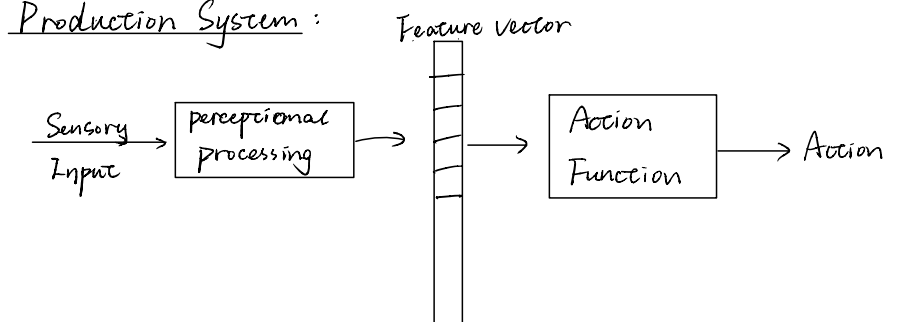
- Copy 10% of the programs from generation n to generation $n+1$. These programs are chosen by the following tournament selection process (Page 33 and 34): 7 programs are randomly selected from the population, and the most fit of these seven is chosen.
- The rest (90%) are produced from generation n by a crossover operation (Page 35) as follows: two parents chosen from generation n by the tournament selection process, and a randomly chosen subtree of one parent is used to replace a randomly selected subtree of another parent.
- Sometimes a mutation operator is also used: it selects a member from generation n by the tournament selection process, and then a randomly chosen subtree of it is deleted and replaced by a random tree. When used, the mutation operation is used sparingly (maybe 1% rate).

State Machine

S-R agent: have no memory

State machine: remember previous action

Production System:



State Machine:

