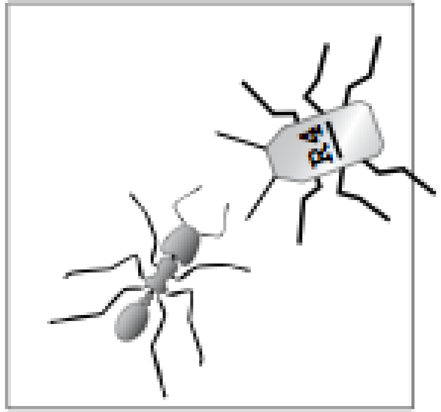
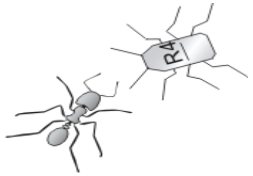
Evolution of Neurocontrollers

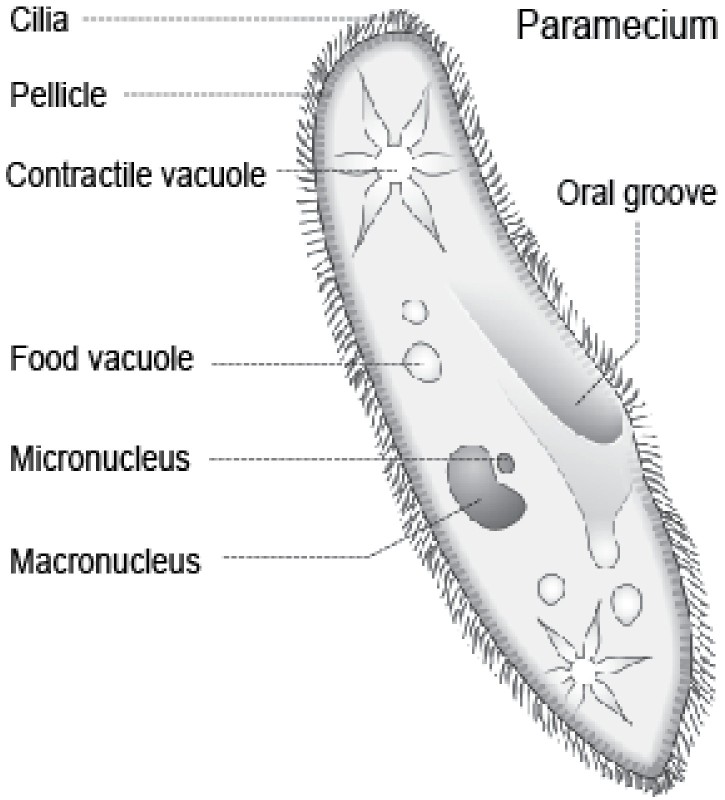


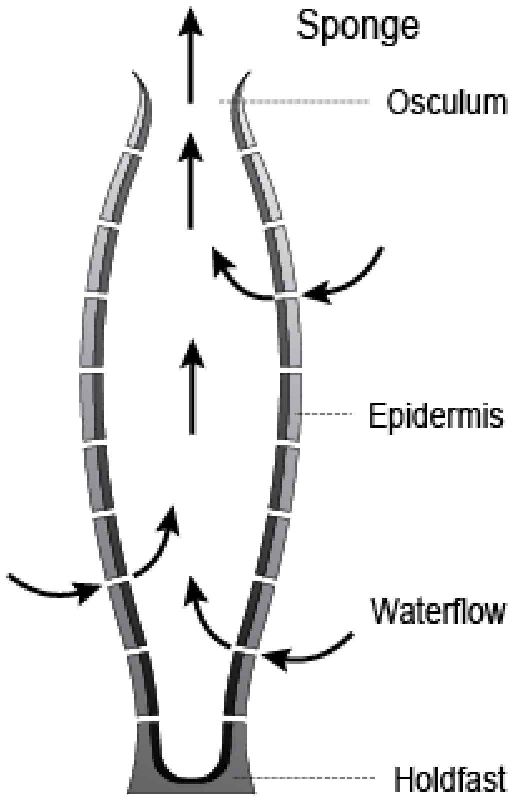
What you will learn in this class

- Neural networks 101
- Genetic encodings of neural controllers
- How to set up, perform, and analyze experiments
- Evolution of vision-based neuro-controllers
- Evolution of spiking neuro-controllers
- Evolution of active perception
- Comparing fitness functions: The Fitness Design Space
- Evolutionary control vs Reinforcement Learning



Do animals need nervous systems?

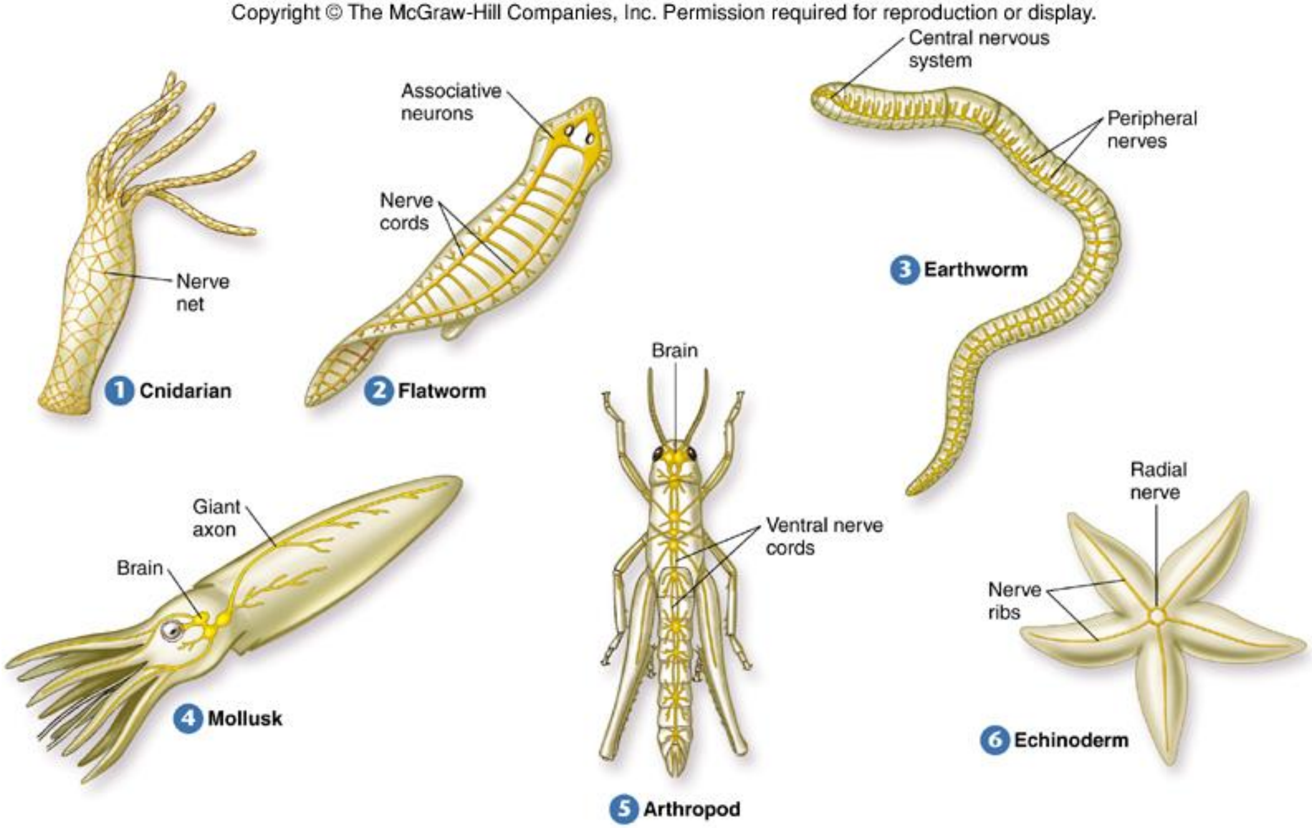


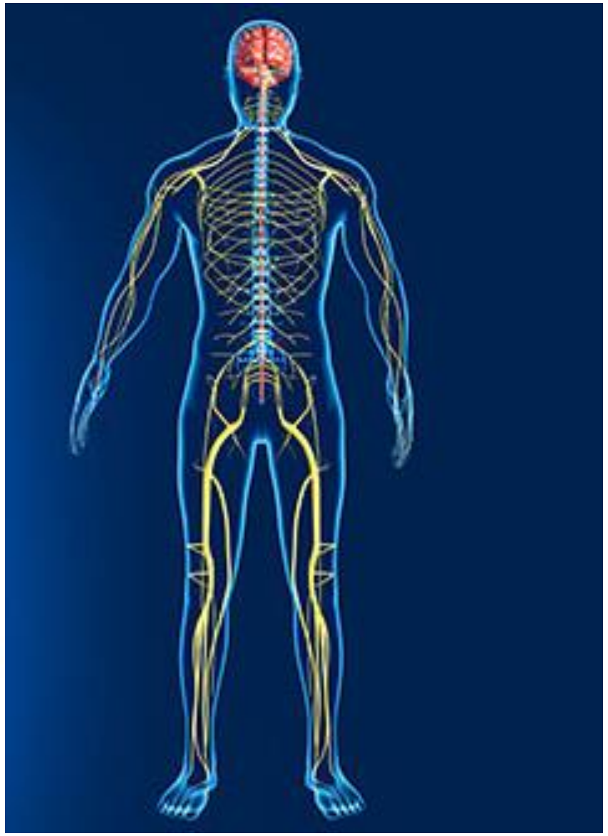


Not all animals have nervous systems; some use only chemical reactions

Paramecium and sponge move, eat, escape, display habituation

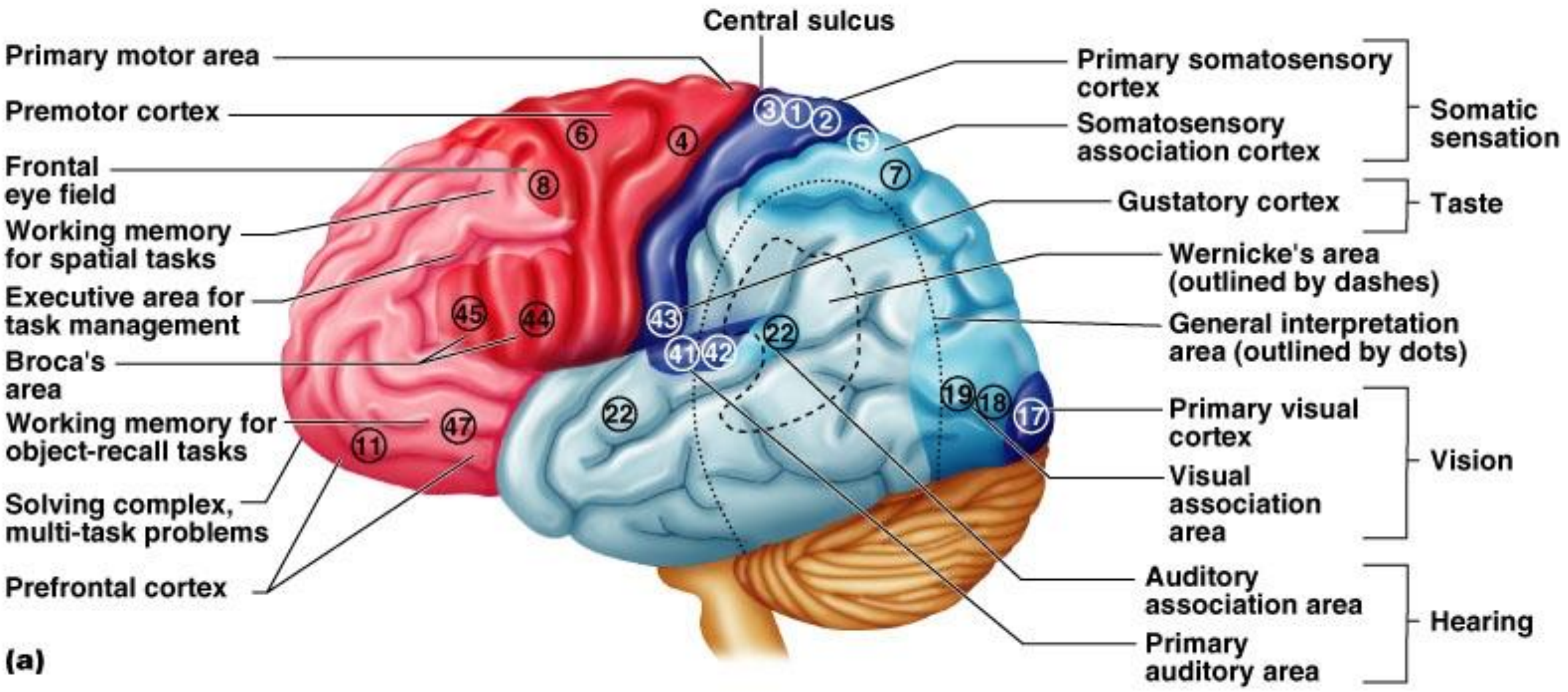
Why Nervous Systems?



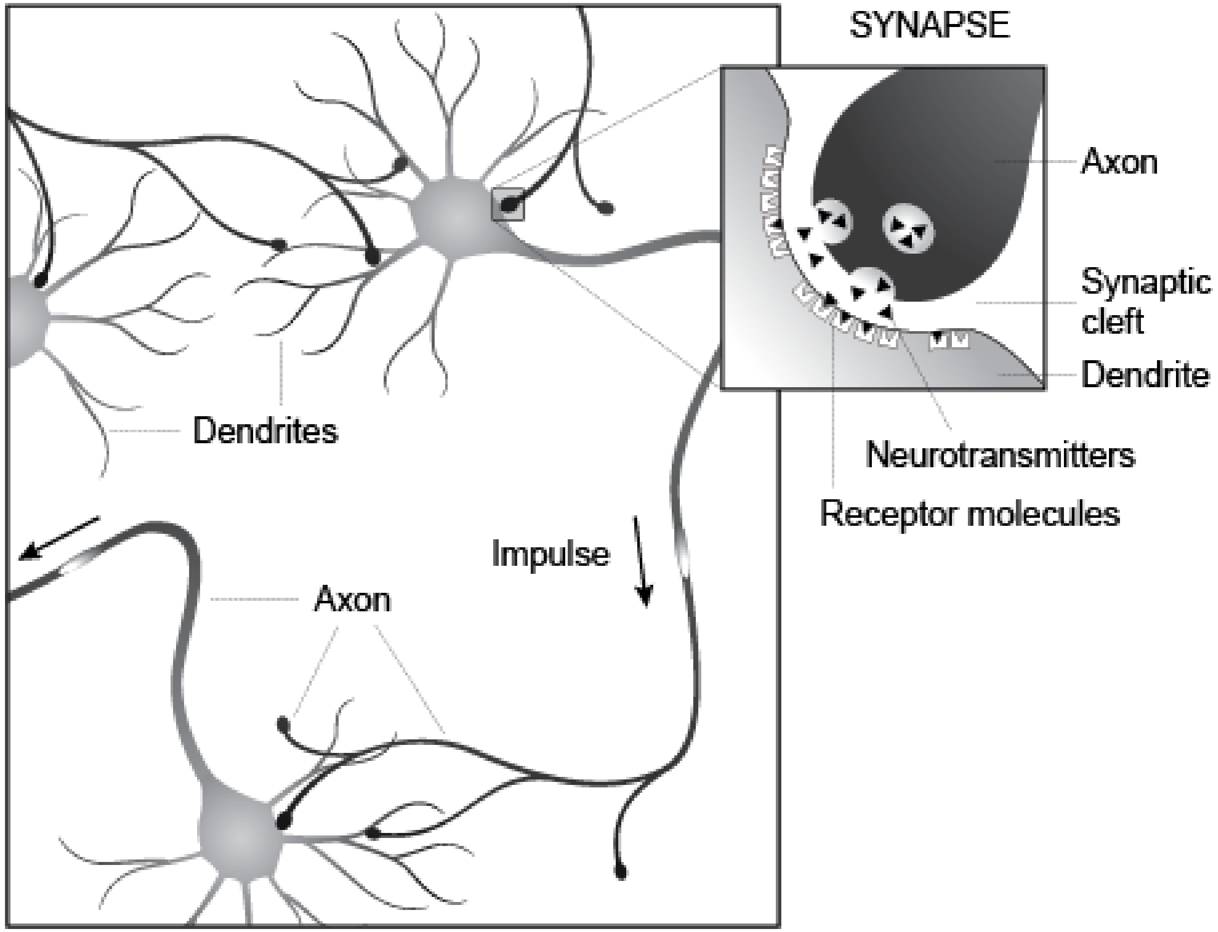


1) Faster reaction times = competitive advantage
2) Selective transmission of signals across distant areas = more complex bodies
3) Generation of non-reactive behaviors
4) Complex adaptation = survival in changing environments

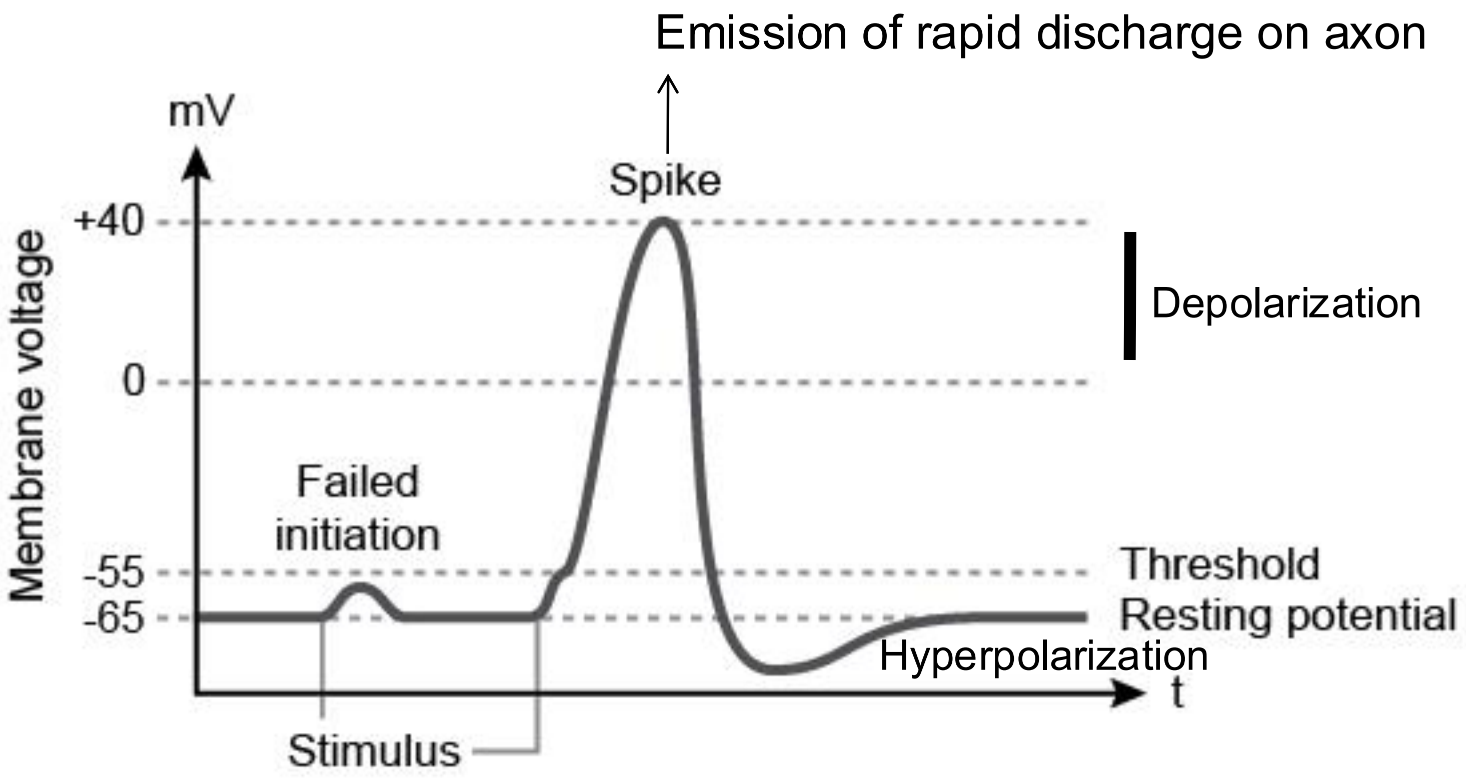
Central Nervous System with Cortex


Central sulcus
Primary motor area
Premotor cortex
Frontal eye field
Working memory for spatial tasks
Executive area for task management
Broca's area
Working memory for object-recall tasks
Solving complex, multi-task problems
Prefrontal cortex
(a)
Primary somatosensory cortex
Somatosensory association cortex
Somatic sensation
Gustatory cortex
Taste
Wernicke's area (outlined by dashes)
General interpretation area (outlined by dots)
Primary visual cortex
Visual association area
Vision
Auditory association area
Primary auditory area
Hearing
3
1
2
5
7
6
4
8
45
44
43
41
42
22
22
47
11
19
18
17

Biological Neurons

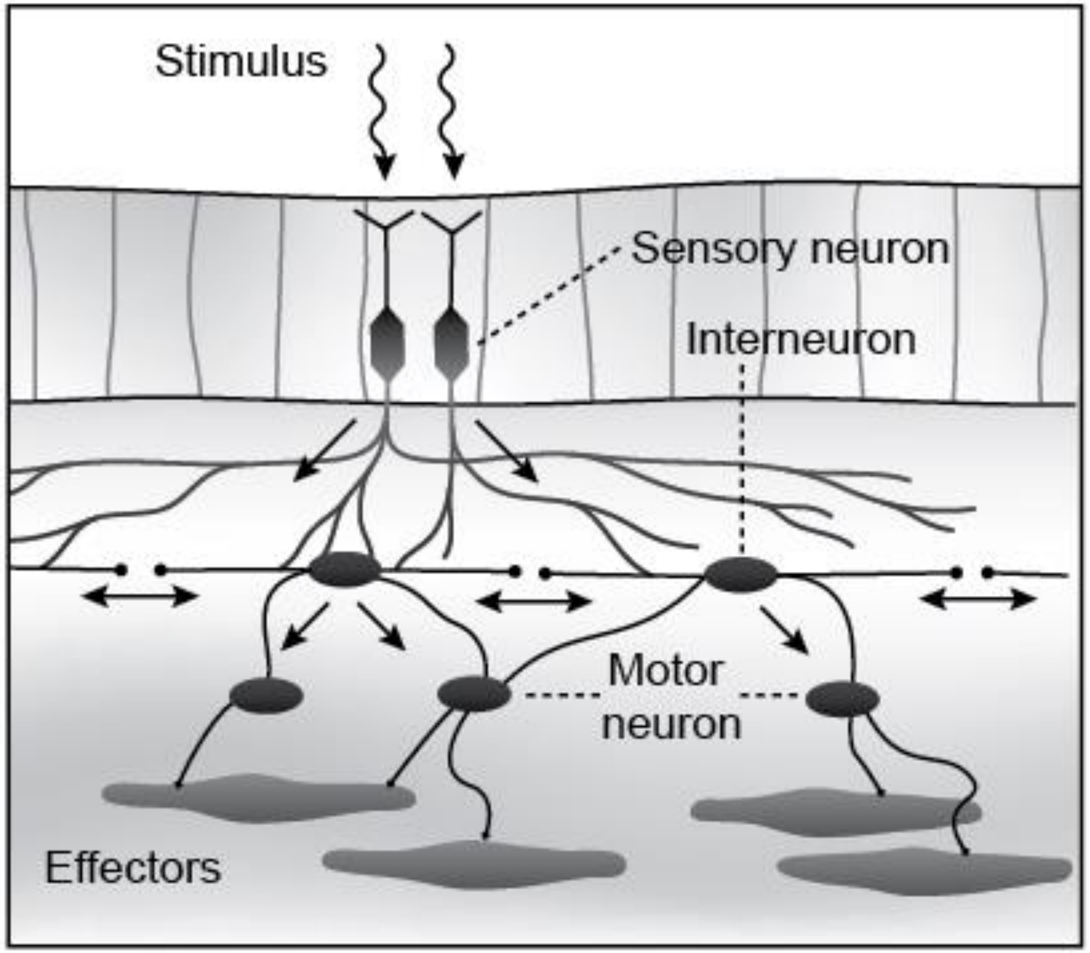


Dynamics of neural activation



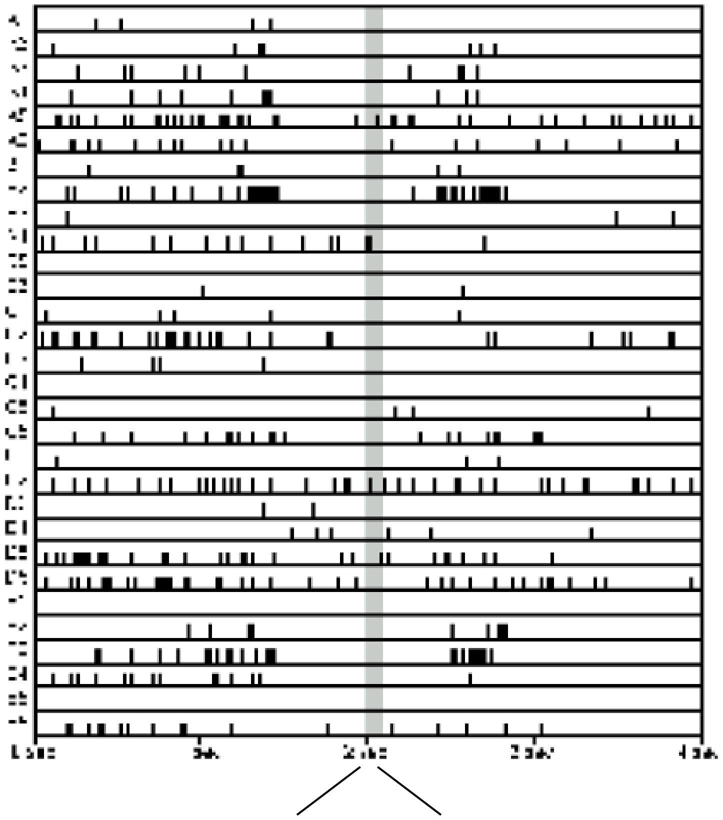
This cycle lasts approximately 3-50 ms, depending on type of ion channels involved (Hodgkin and Huxley, 1952)

Types of Neurons

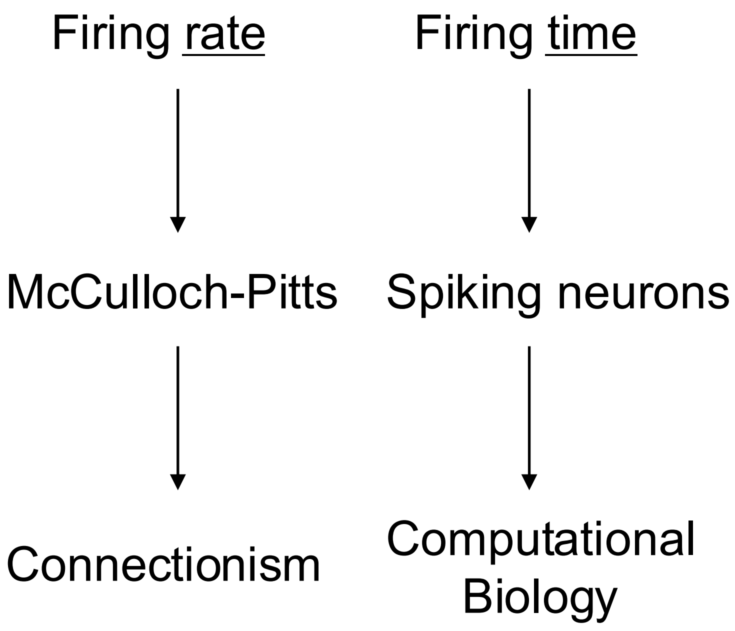


Interneurons can be
1- Excitatory
2- Inhibitory

How Do Neurons Communicate?



100 ms



Firing rate

McCulloch-Pitts

Connectionism

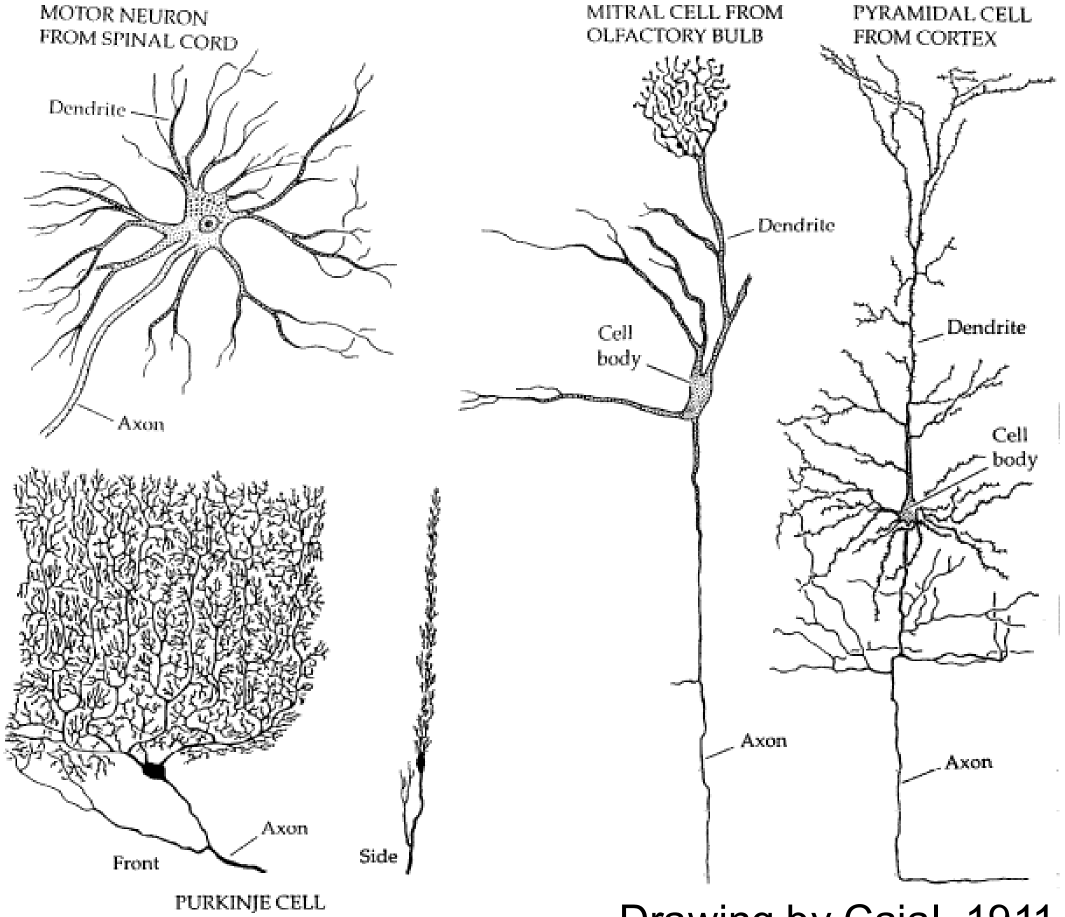
Firing time

Spiking neurons

Computational Biology

What Does Make Brains Different?

Components and behavior of individual neurons are very similar across animal species and, presumably, over evolutionary history (Parker, 1919)

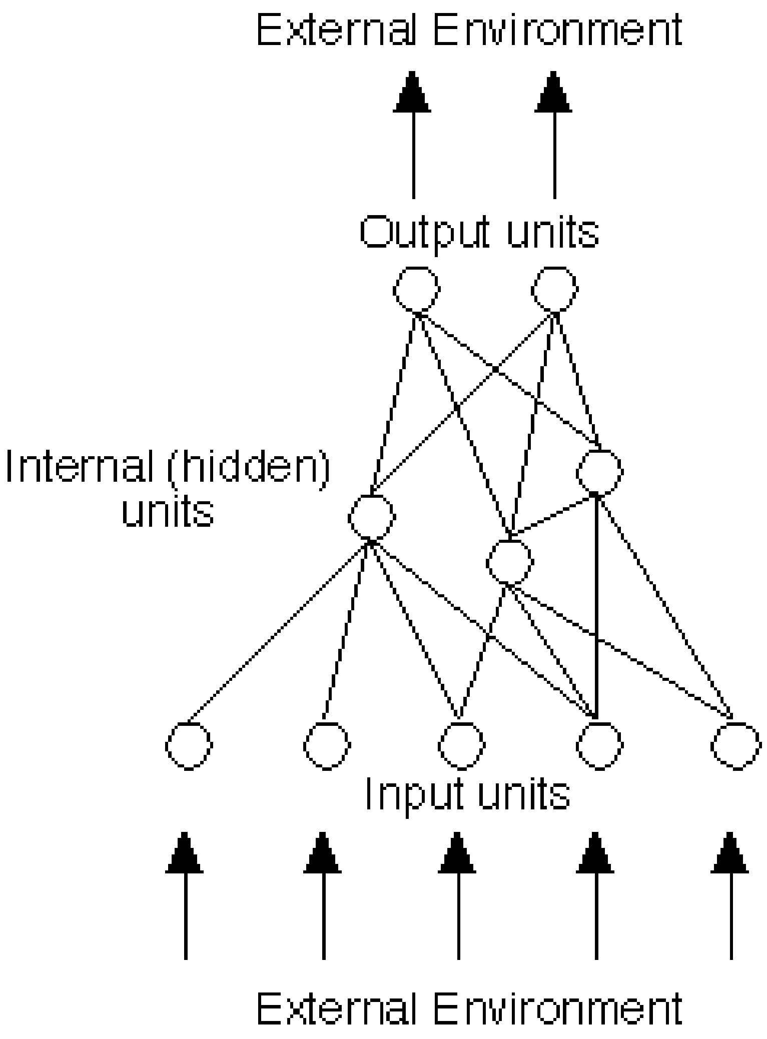


Drawing by Cajal, 1911

Evolution of the brain seems to occur mainly in the **architecture**, that is how neurons are interconnected.

First classification of neurons by Cajal in 1911 was made according to their connectivity patterns

An Artificial Neural Network



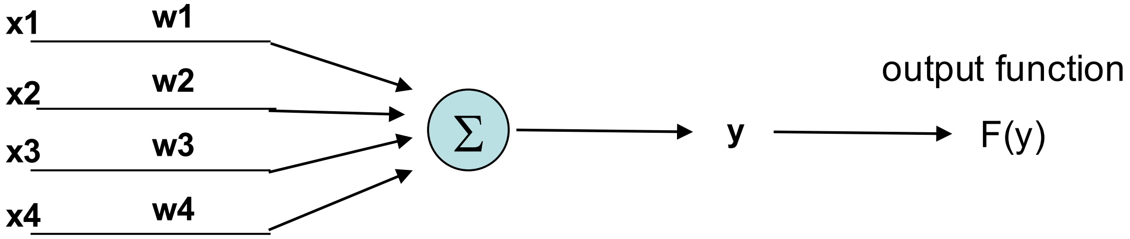
A neural network communicates with the environments through input units and output units. All other elements are called internal or hidden units.

Units are linked by uni-directional connections.

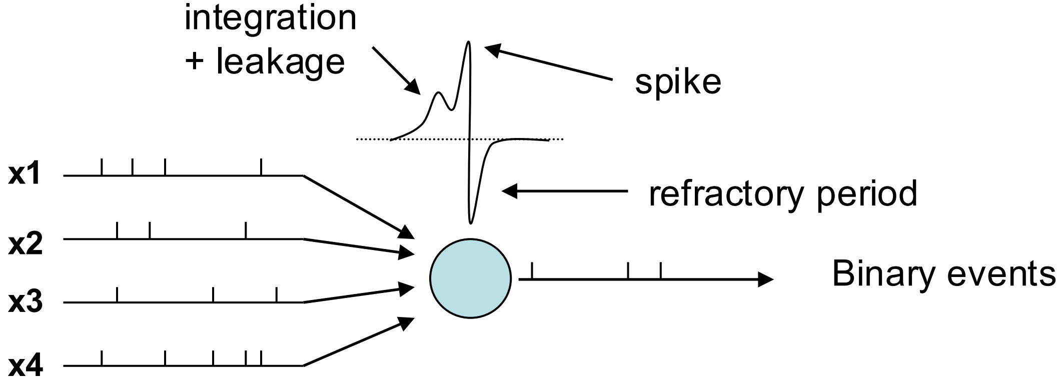
A connection is characterized by a weight and a sign that transforms the signal.

Neuron models

McCulloch-Pitts

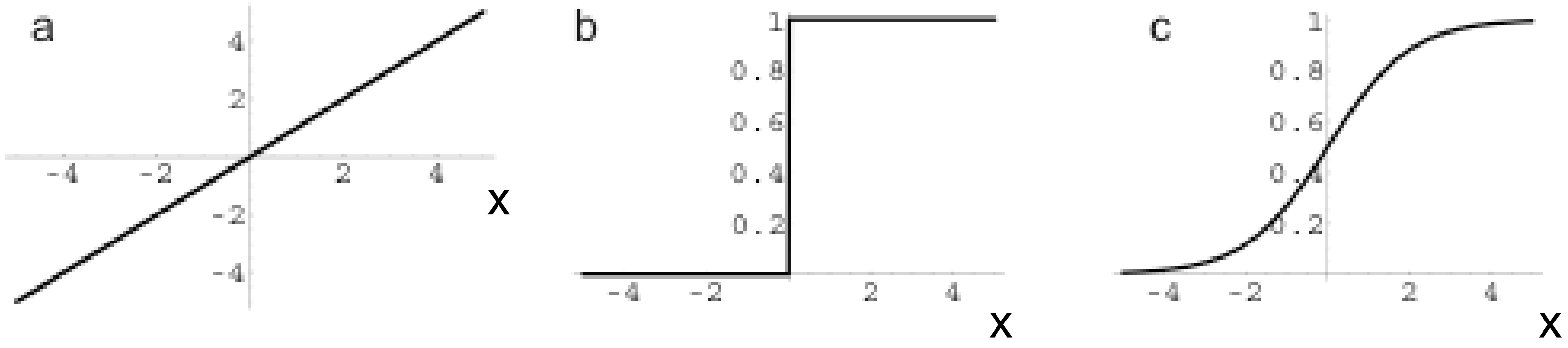


Spiking



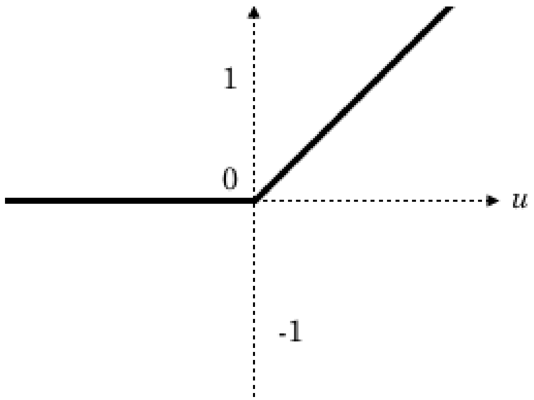
Some output functions

Linear $\Phi(x)$ Step $\Phi(x)$ Sigmoid $\Phi(x)$



Rectified Linear

$f(u) = \max(0, u)$



Sigmoid function:
- continuous
- non-linear
- monotonic
- bounded
- asymptotic

$$\Phi(x) = \frac{1}{1+e^{-kx}}$$

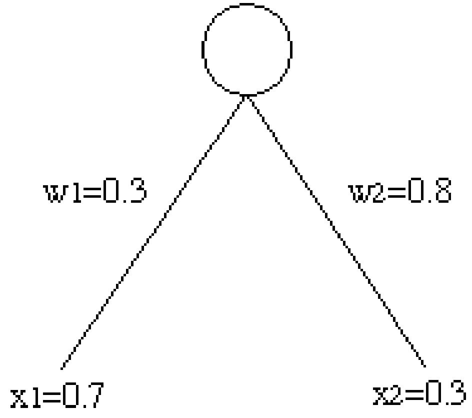
$$\Phi(x) = \tanh(kx)$$

Understanding single neuron computation

The output of a neuron is a measure of similarity between its input vector and its weight vector

1. Output of a neuron is the dot product of the weight and input vectors:

$$y = a\left(\sum_{i}^{N} w_i x_i\right), \qquad a = 1 \longrightarrow y = \mathbf{w} \cdot \mathbf{x}$$



2. Distance between two vectors is:

$$\cos \vartheta = \frac{\mathbf{w} \cdot \mathbf{x}}{\|\mathbf{w}\| \|\mathbf{x}\|}, \qquad 0 \leq \vartheta \leq \pi$$

where the vector length is:

$$\|\mathbf{x}\| = \sqrt{\mathbf{x} \cdot \mathbf{x}} = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$$

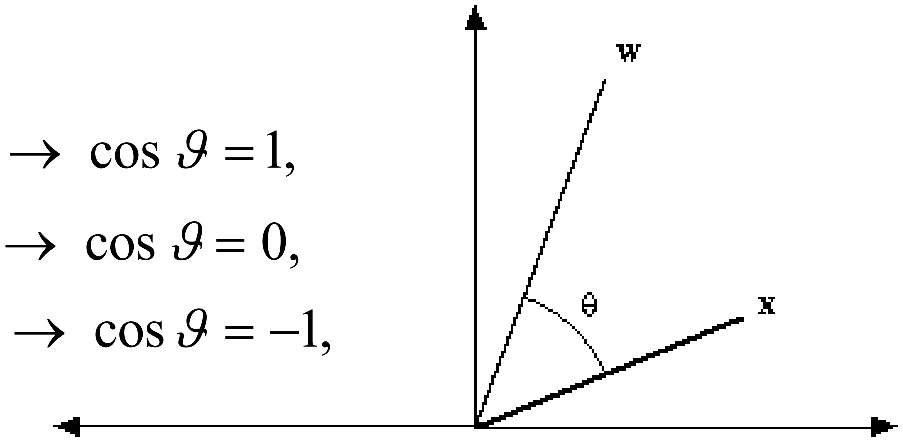
3. Output signals vector distance (familiarity)

$$\mathbf{w} \cdot \mathbf{x} = \|\mathbf{w}\| \|\mathbf{x}\| \cos \vartheta$$

$$\vartheta = 0^{\mathrm{o}} \rightarrow \cos \vartheta = 1,$$

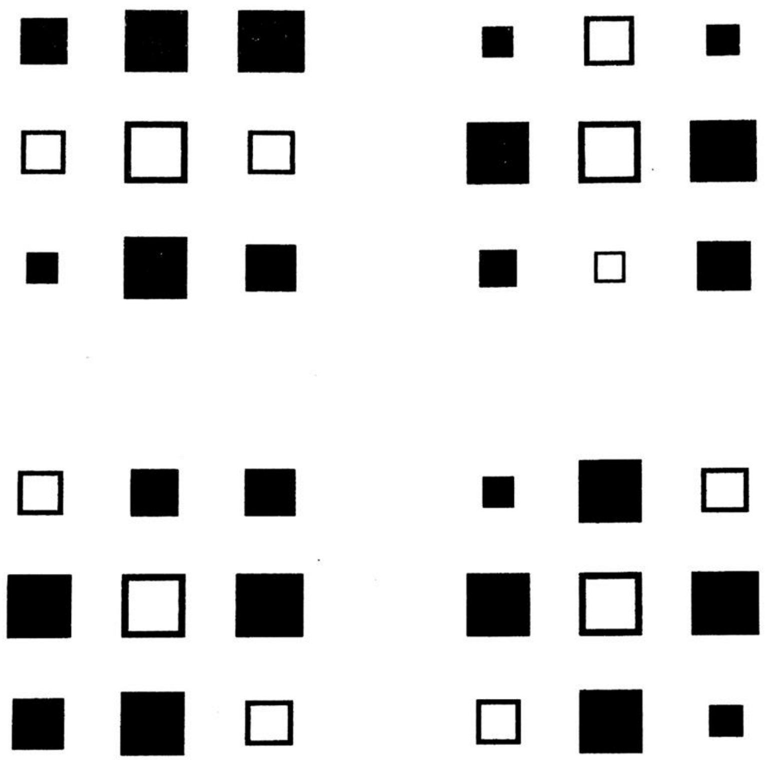
$$\vartheta = 90^{\mathrm{o}} \rightarrow \cos \vartheta = 0,$$

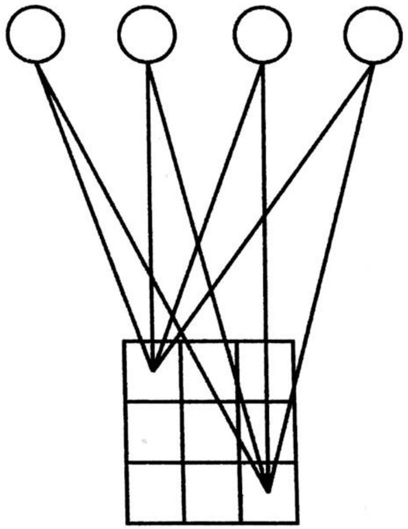
$$\vartheta = 180^{\mathrm{o}} \rightarrow \cos \vartheta = -1,$$



Neural Receptive Fields

The **Receptive Field** indicates the input area subtended by a neuron *and* the input pattern that generates the strongest activation.
RF can be visualized by plotting the weight pattern in the input space.



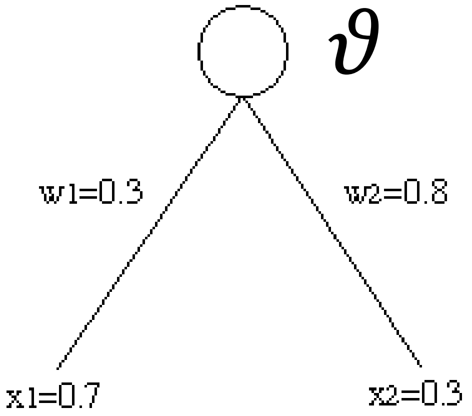


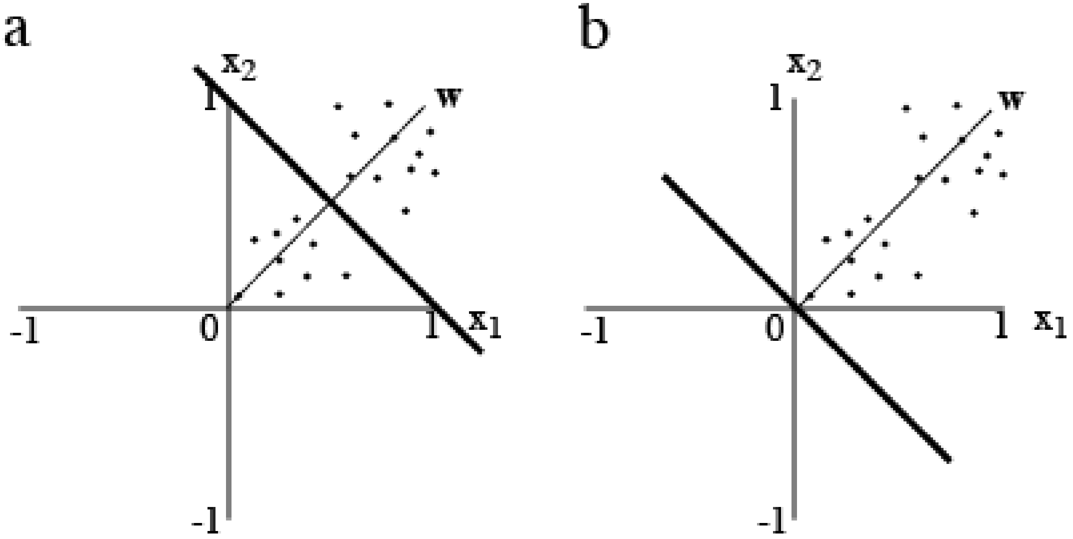
Fully connected
(only some connections are shown)

Neurons as classifiers

A neuron divides the input space in two regions, one where weighted input sum >=0 and one where weighted input sum <0. The separation line is defined by the synaptic weights

$$w_1 x_1 + w_2 x_2 - \vartheta = 0 \qquad x_2 = \frac{\vartheta}{w_2} - \frac{w_1}{w_2} x_1$$





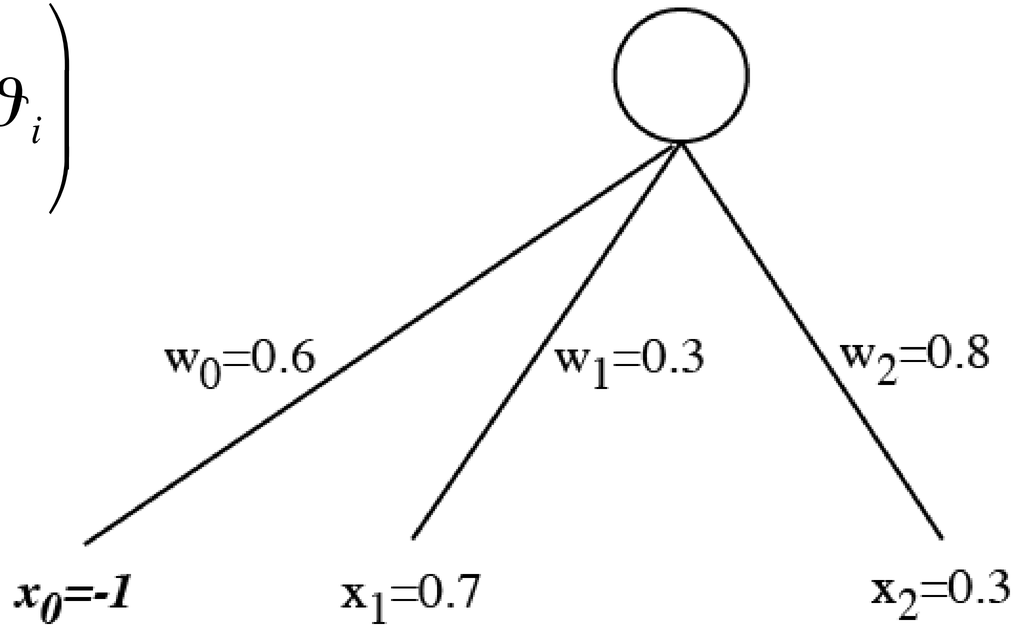
$\vartheta > 0$ $\qquad$ $\vartheta = 0$

From Threshold to Bias unit

The threshold can be expressed as an additional weighted input from a special unit, known as bias unit, whose output is always -1.

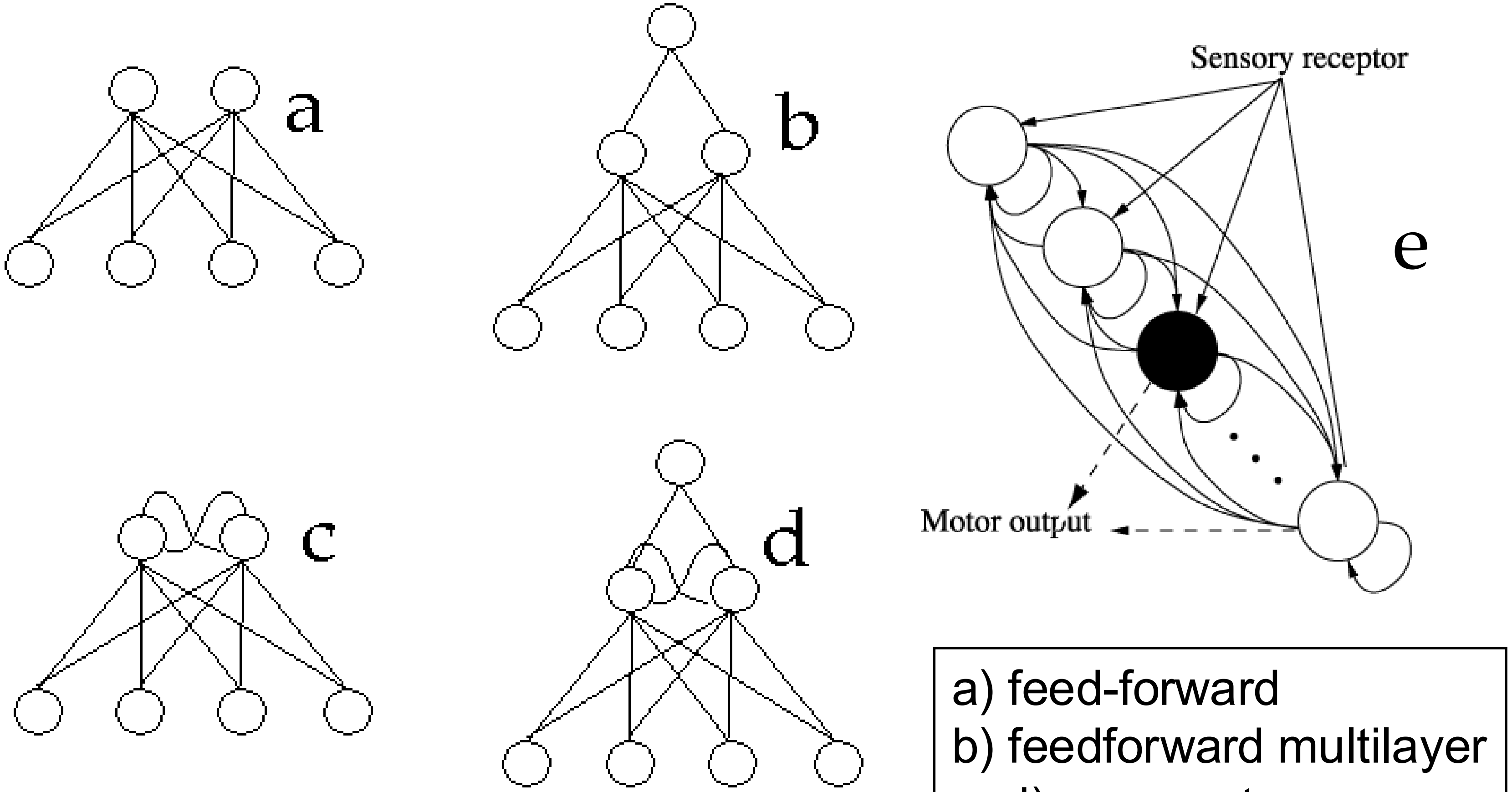
$$y_i = \Phi(A_i) = \Phi\left(\sum_{j=1}^{N} w_{ij}x_j - \vartheta_i\right)$$

$$y_i = \Phi(A_i) = \Phi\left(\sum_{j=0}^{N} w_{ij}x_j\right)$$



- Easier to express/program
- Threshold is adaptable like other weights

Architectures

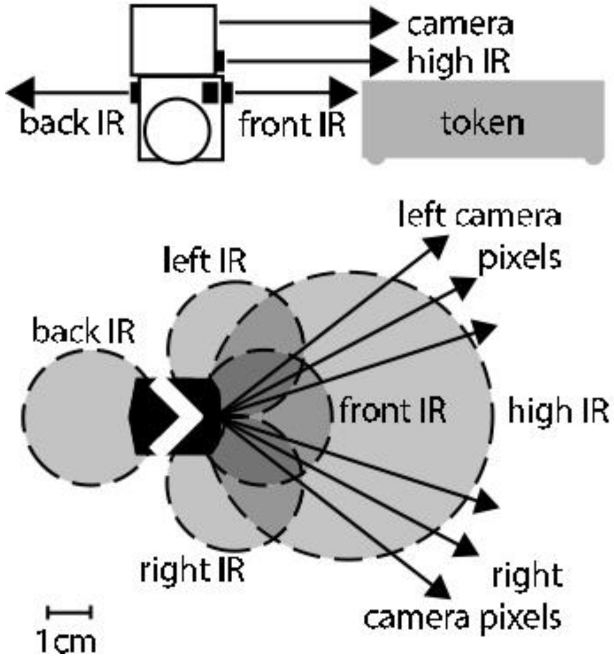


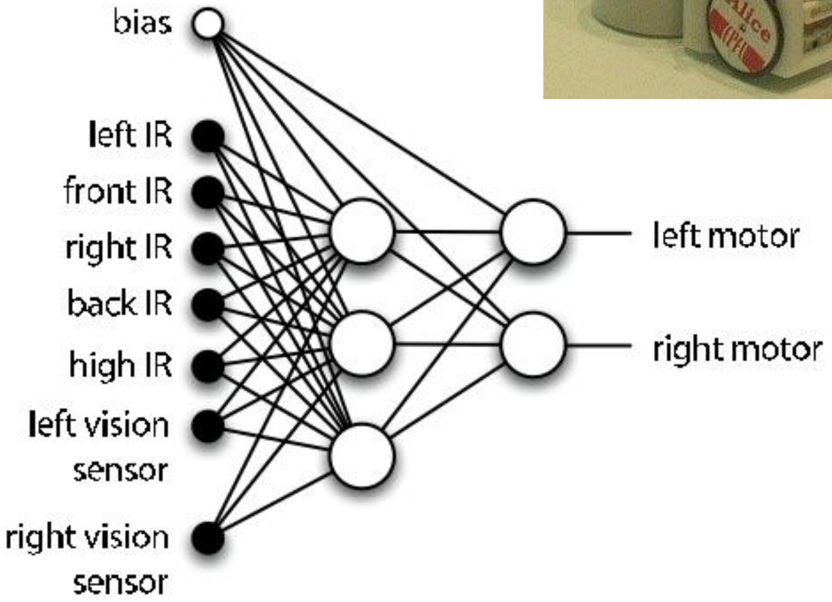
a) feed-forward
b) feedforward multilayer
c, d) recurrent
e) fully connected

Input normalisation to give all sensors equal relevance

Input signals from different sensory sources can have different amplitudes that must be normalised to enable comparisons by receiving neurons

$$x'_i = \frac{x_i}{\sqrt{\sum_{j=1}^{N} x_j^2}}$$





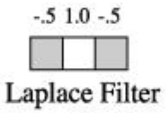
Convolution to encode features

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

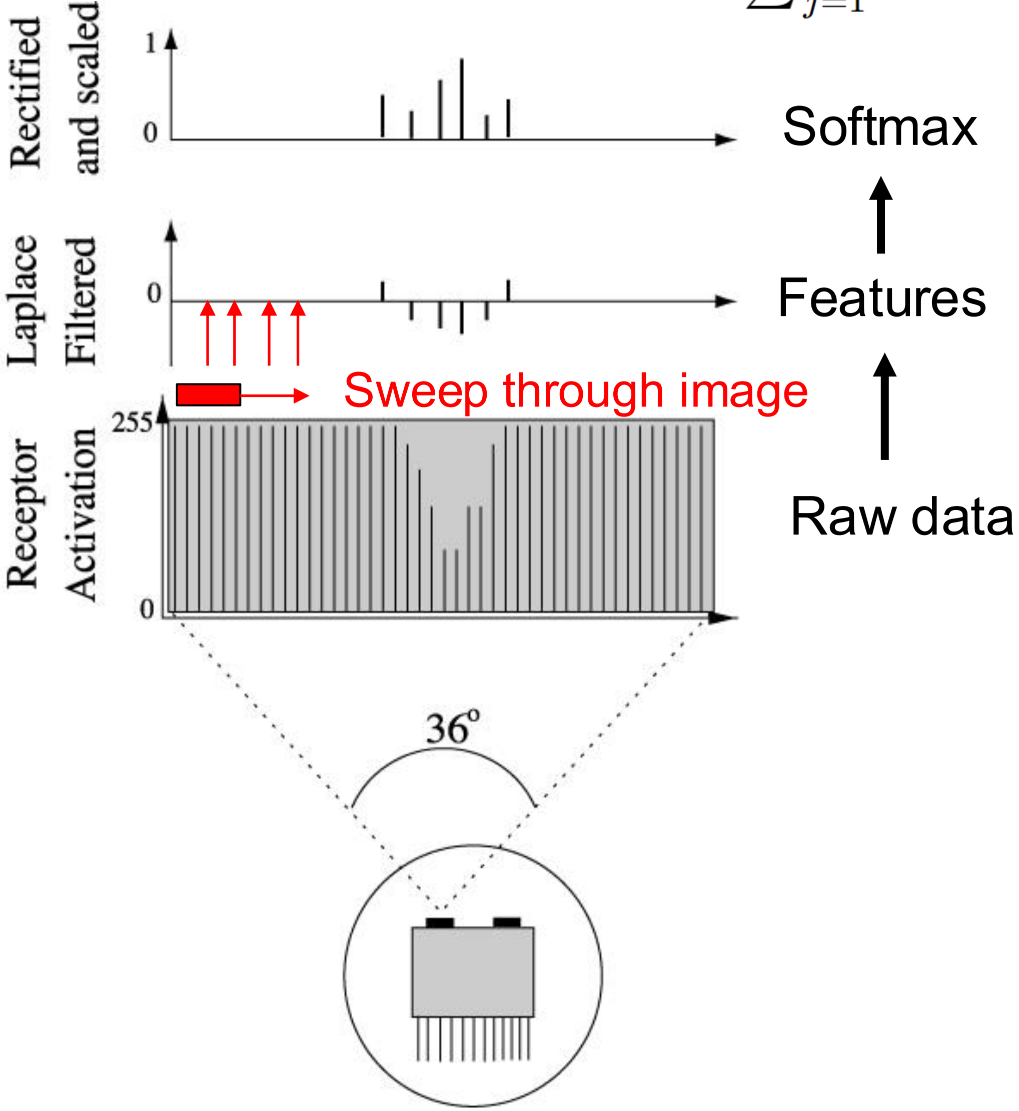
We are interested in features, not pixels

Feature detection by image convolution with filters

- Predesigned filters (e.g., Laplace filter)



- Learned filters (autoencoders, etc.)
- Evolved filters



Softmax

Features

Sweep through image

Raw data

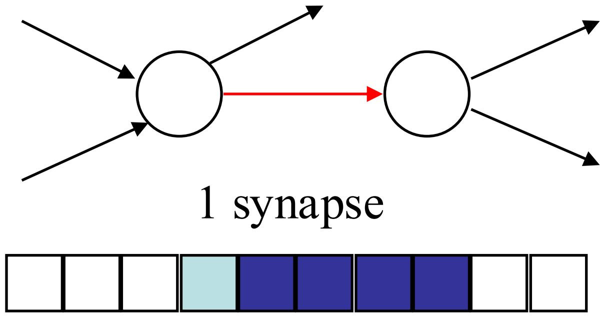
Genome can encode

1. Connection Weights
   a. pre-defined neural network architecture
   b. binary or real-valued representation of connection weights
   c. fixed-length genotype

2. Learning Rules
   a. pre-defined neural network architecture
   b. Binary or real-valued representation of learning rule
   c. Fixed-length genotype

3. Topology
   a. Neural network architecture created at birth
   b. Genotype encodes the parameters of a generative algorithm (program, L-System, neural network)
   c. Fixed-length or variable-length genotype

Evolution of connection weights



Binary encoding

synapse sign

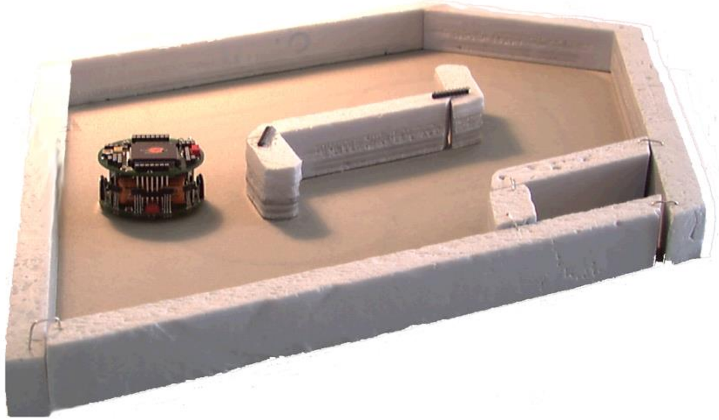
synapse strength

Fitness function is a measure of the robot behavior

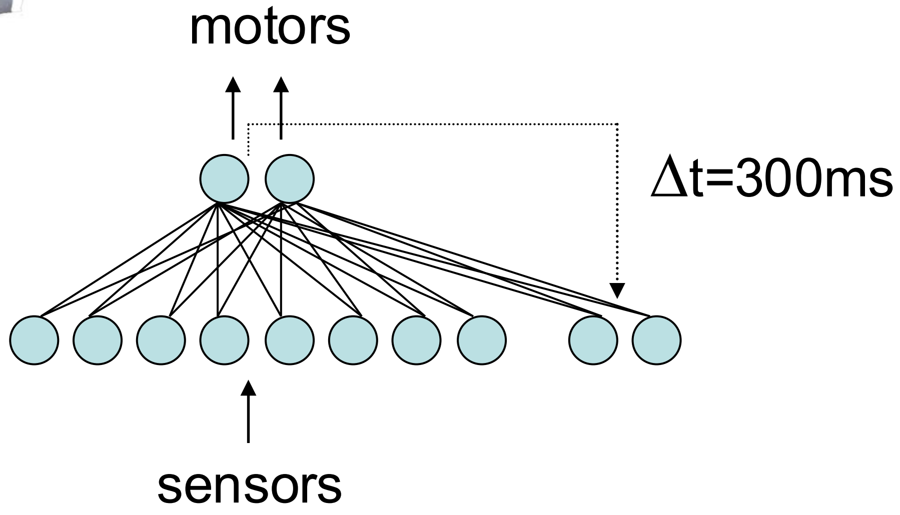
Can be combined with learning:

- learning starts from genetically encoded weights
- fitness measures performance of network after training
- learned weights are not written back into genome

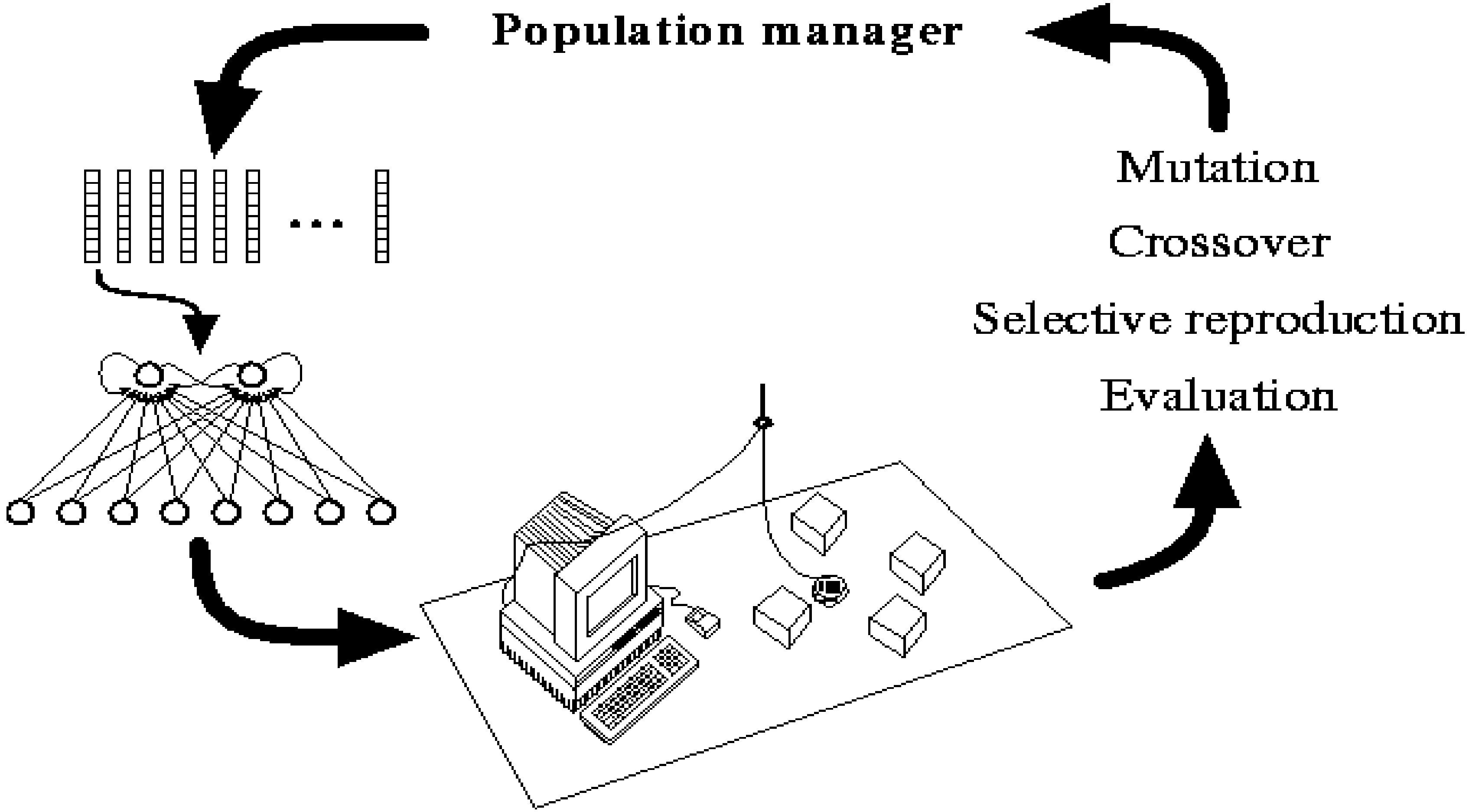
Collision-free Navigation

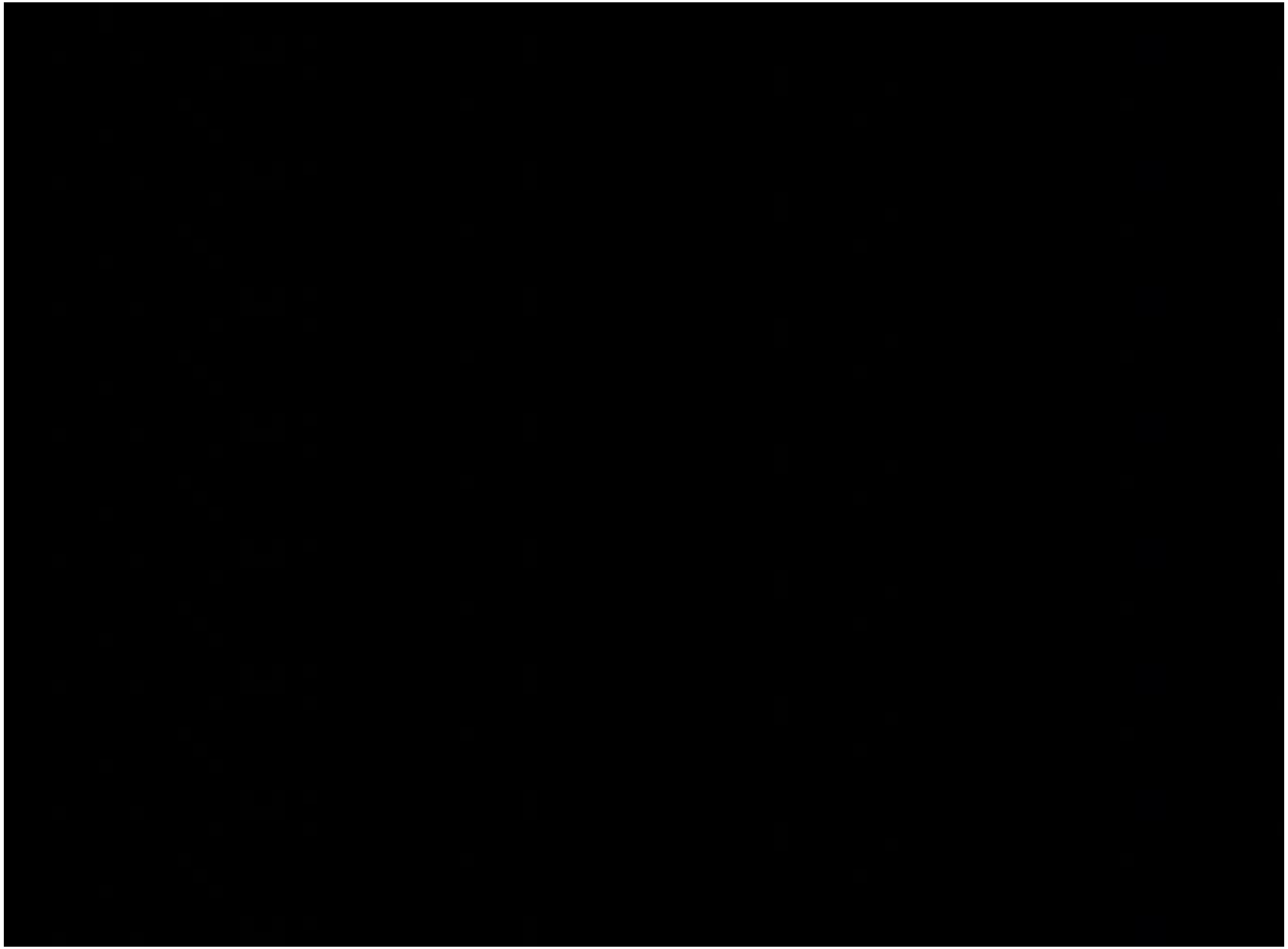


$$\text{Fitness} = V \times \Delta v \times (1-s)$$

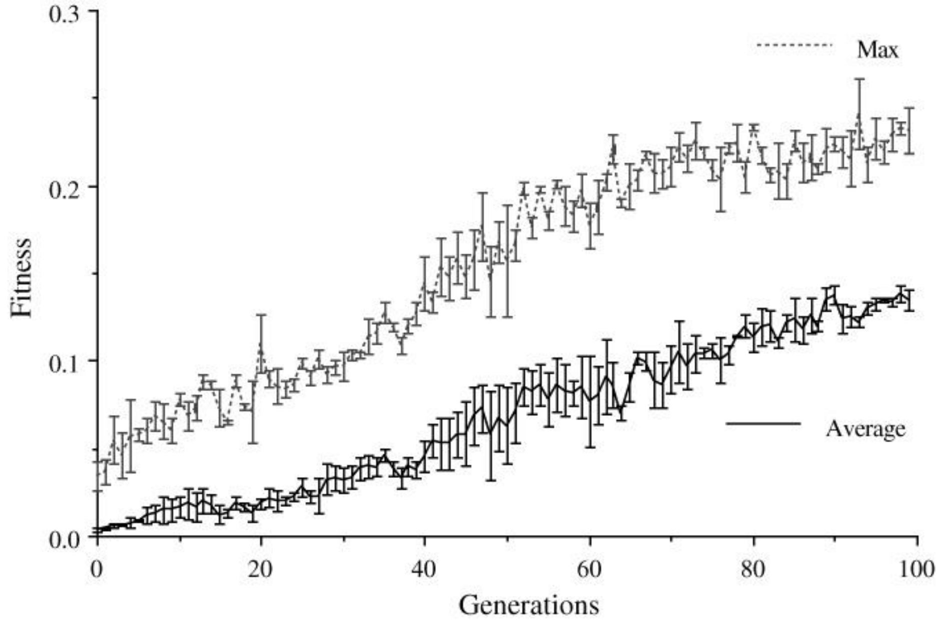


Methodology

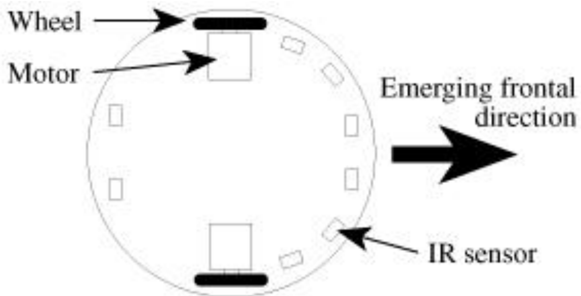




Results



Direction



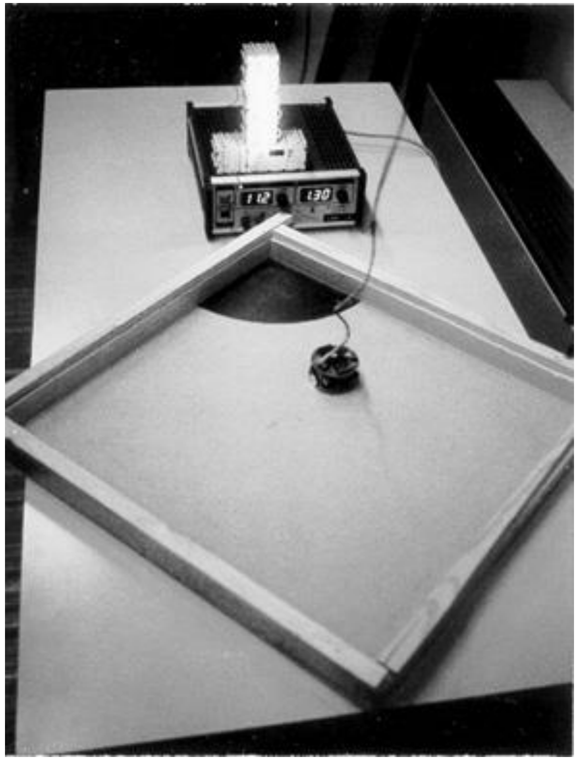
Speed = 60%

The average and best population fitness are typical measures of performance.

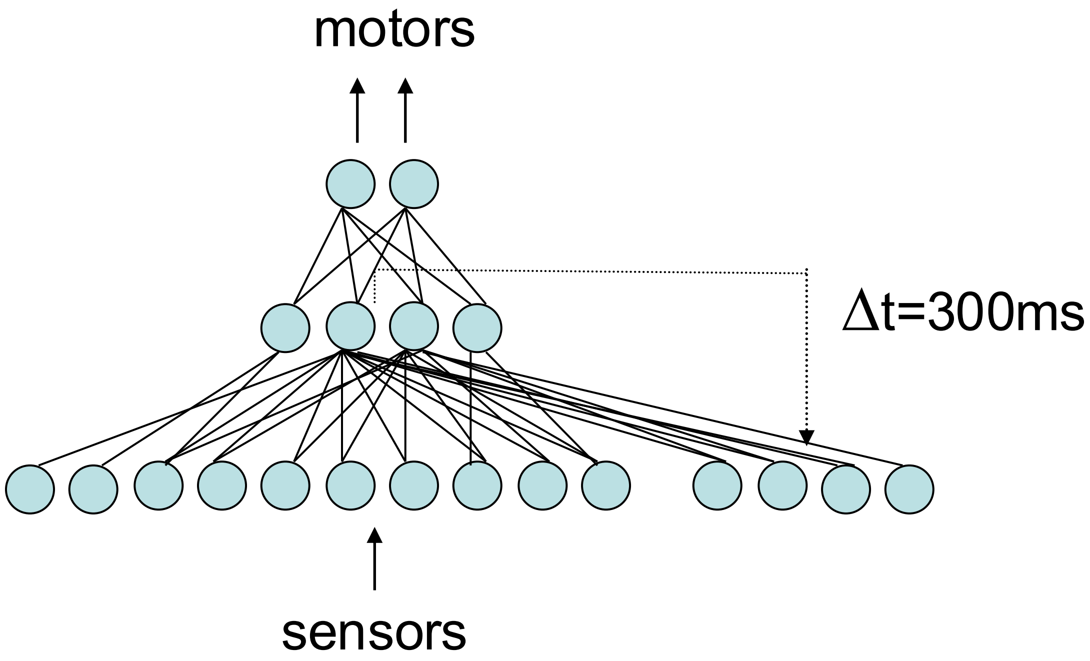
Evolved robots always have a preferential direction of motion and speed.

Homing for Battery Charge

Let us now put the robot in a more complex environment and make the fitness function even simpler. The robot is equipped with a battery that lasts only 20 s and there is a battery charger in the arena.

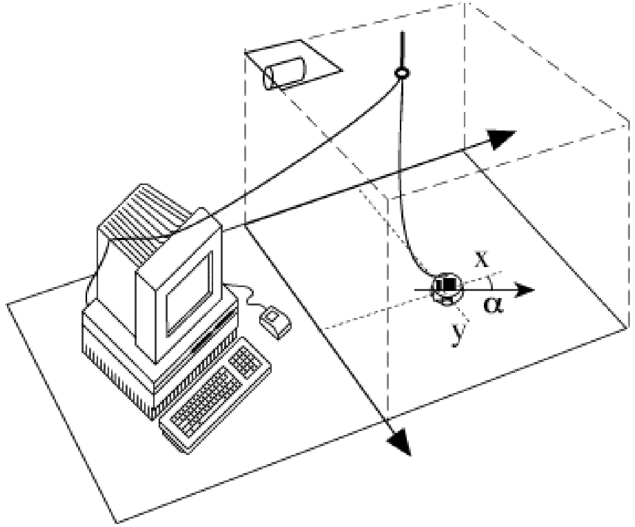


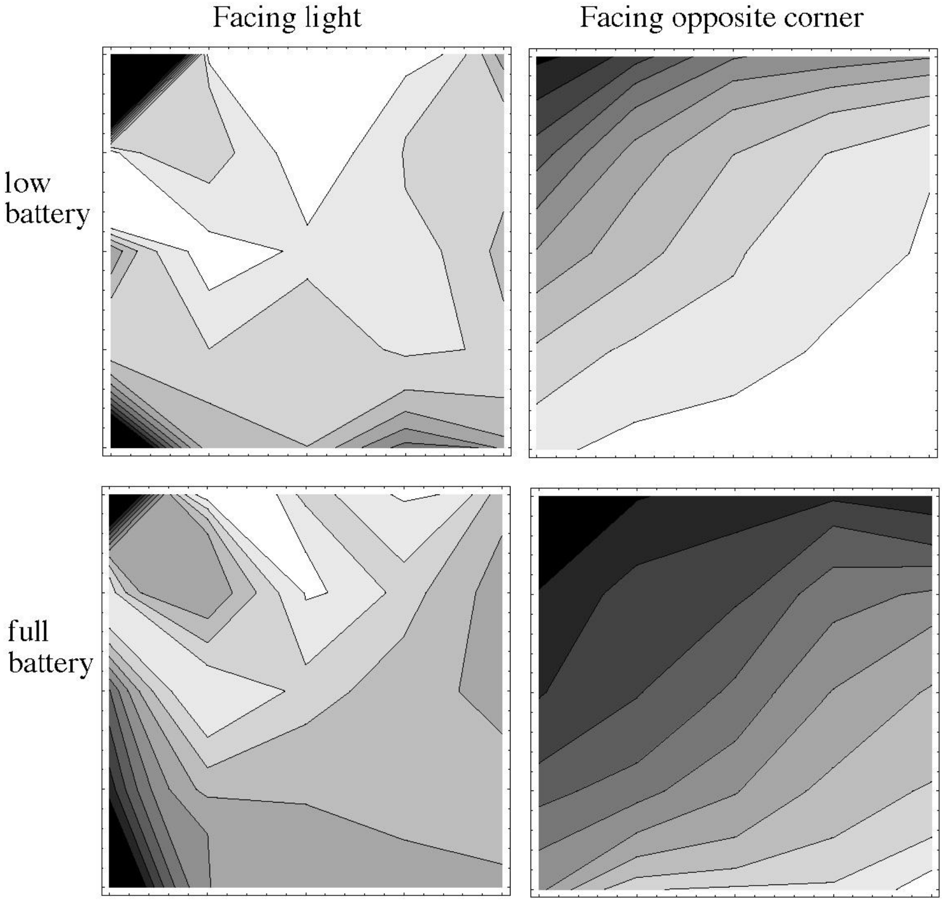
Fitness = V x (1-s)



Machine Neuro-Ethology

Best evolved robots go to recharge with only 10% residual energy. Why and how?

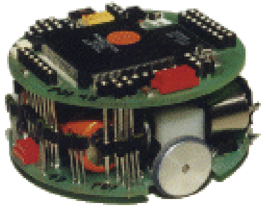




Activity of an internal neuron

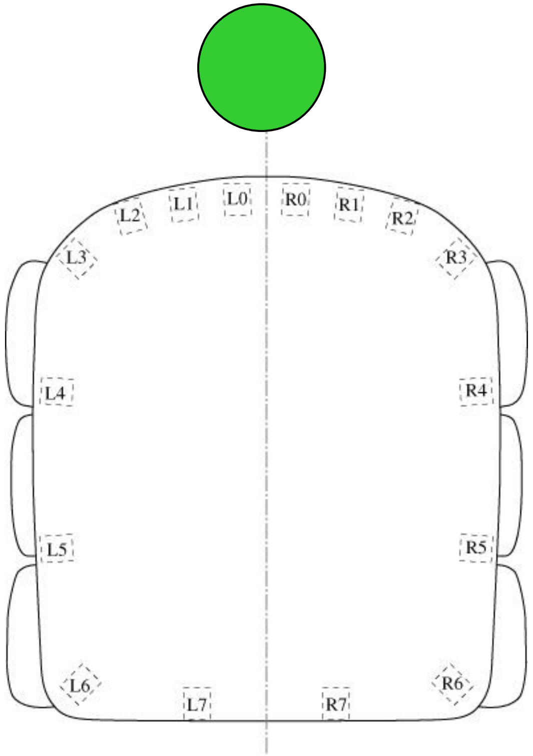
Evolution of complex robots

It is difficult to evolve from scratch large and complex robots because of:
- hardware robustness
- *bootstrap problem:* zero-fitness of all individuals of the initial generation



Khepera robot

Koala robot

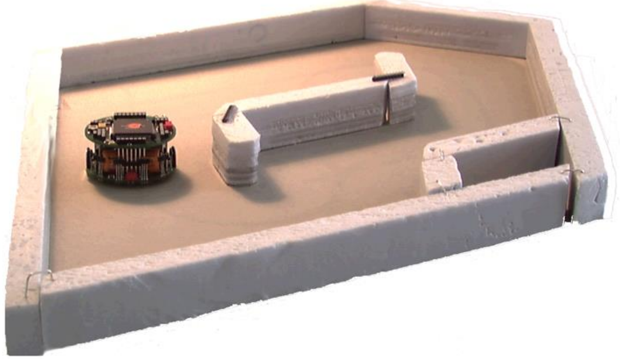


Incremental evolution (a.k.a robot shaping)

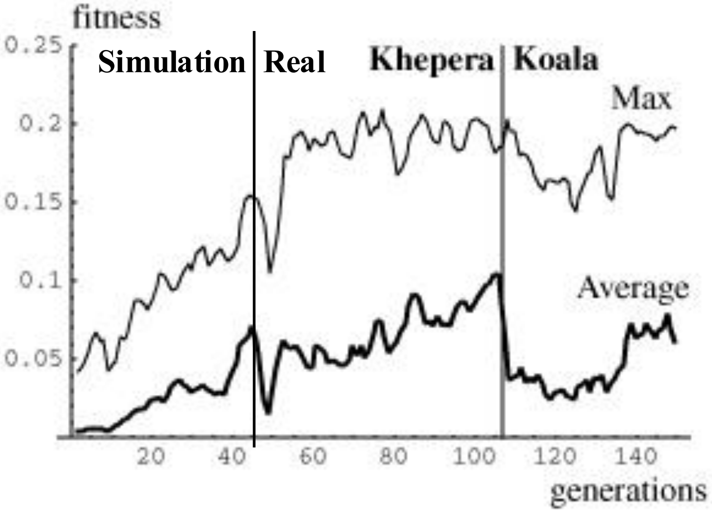
simulation

$Fitness=V \times \Delta v \times (1-s)$

real robot (Khepera)



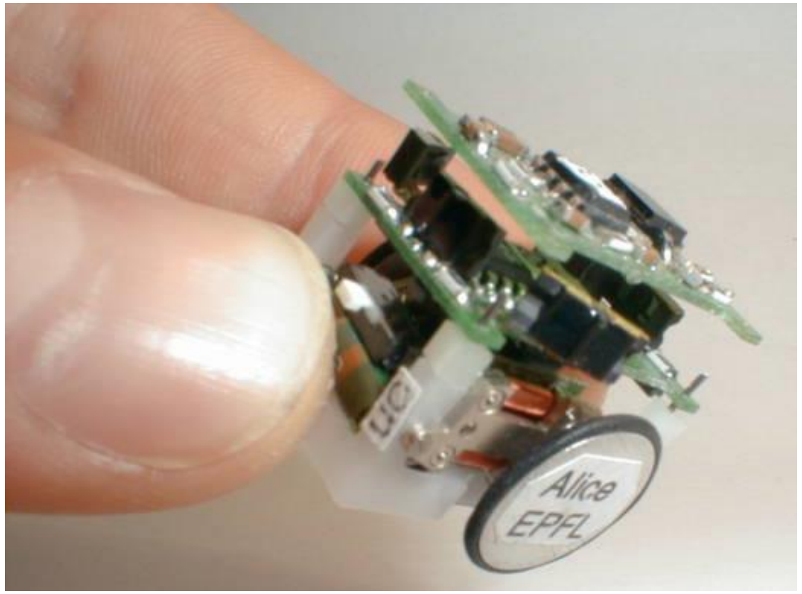
$Fitness=V \times \Delta v \times (1-s)$

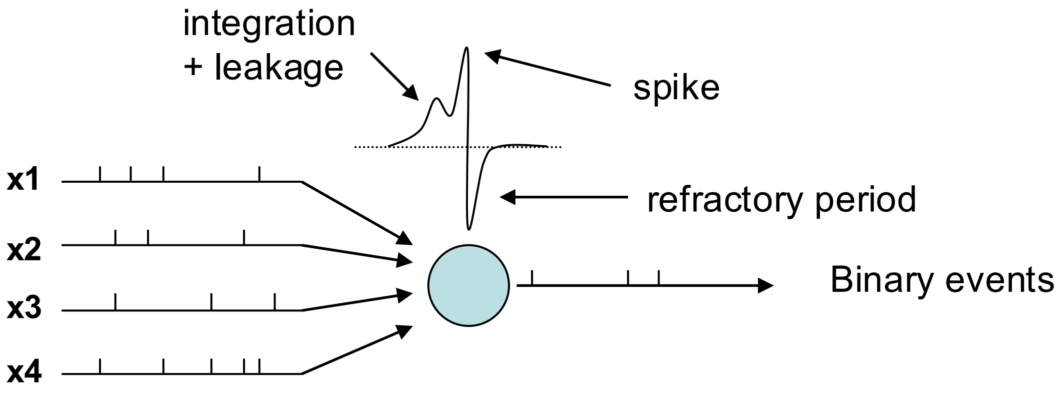


different robot (Koala)

$Fitness=V \times \Delta v \times (1-s)$

Evolution of spiking neural controllers





EPFL Microrobot
- 4 proximity sensors
- 2 Swatch motors
- 10 hours autonomy

Microcontroller PIC16F84, (Microchip, 2001)
1024 words of program memory
68 bytes of RAM
64 bytes of EEPROM

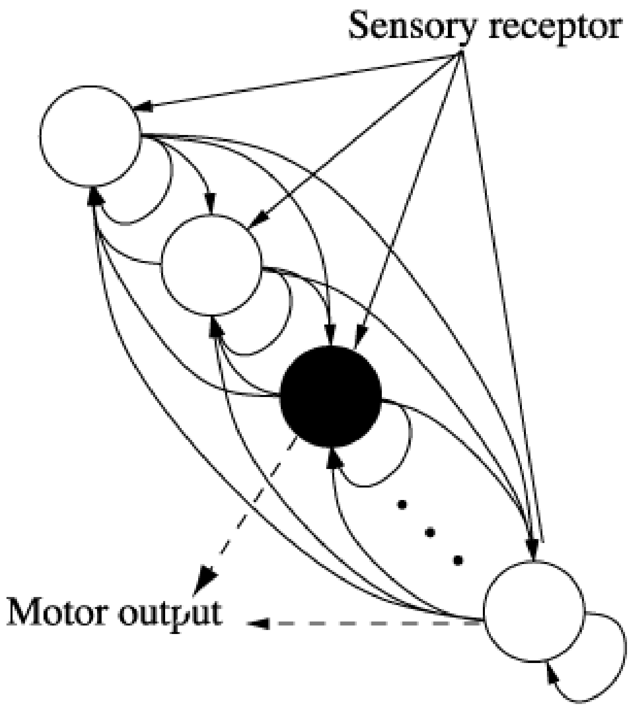
Representation and encoding of neural architecture

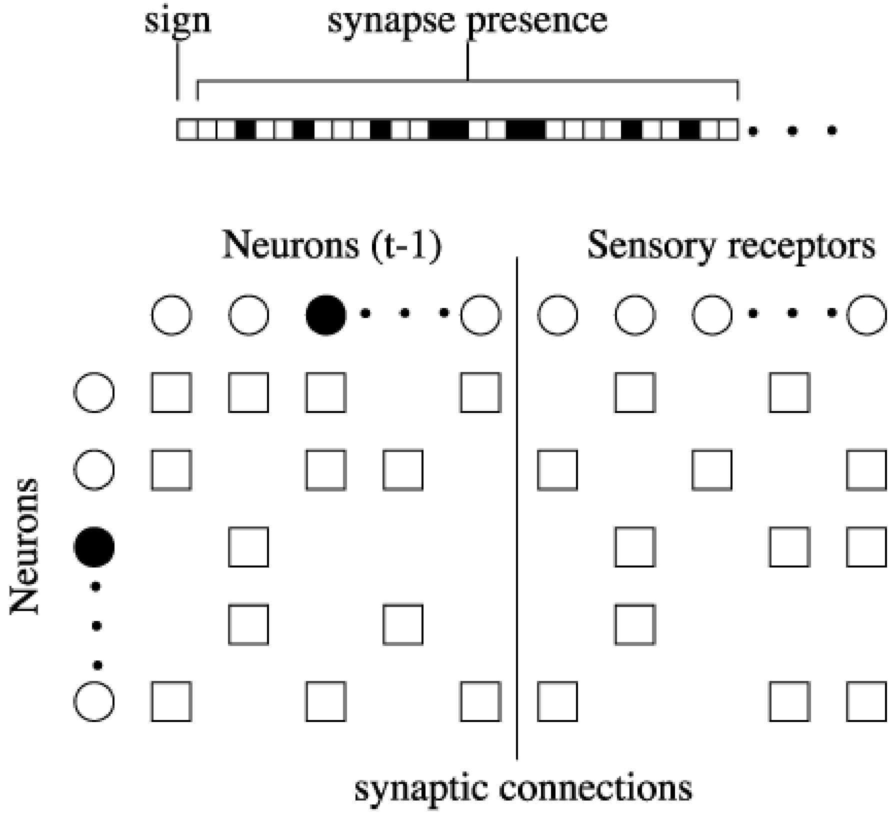
Spring Neurocontroller = 8 fully connected neurons with binary connections (present/absent connection)

Sign of each neuron: excitatory or inhibitory (1bit x 8 neurons = 8 bits)

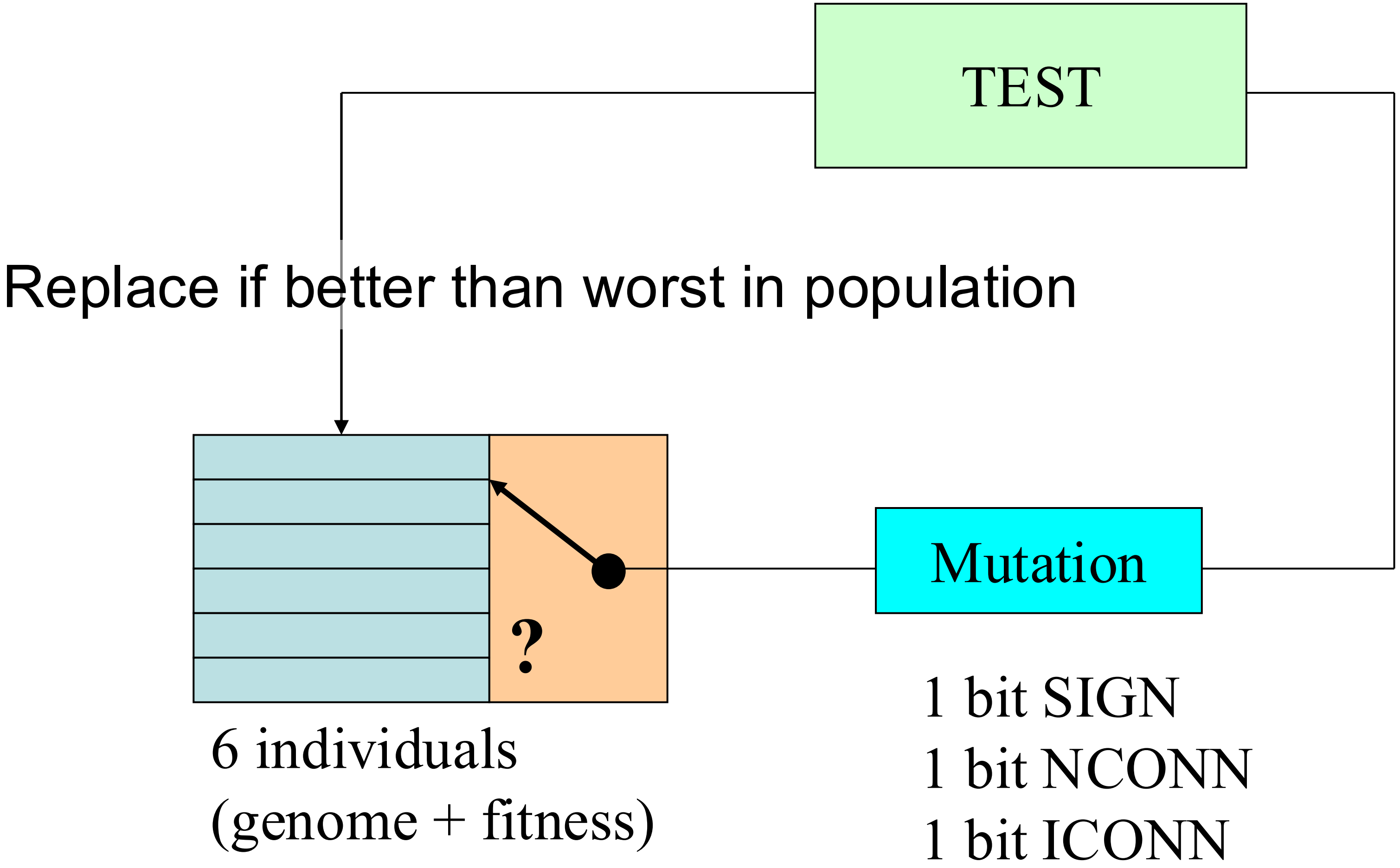
Incoming connection pattern of a neuron: internal connections (8 bits) + sensory connections (8 bits)

Full genome of one neurocontroller = 17 bytes





Steady-state evolutionary algorithm



TEST

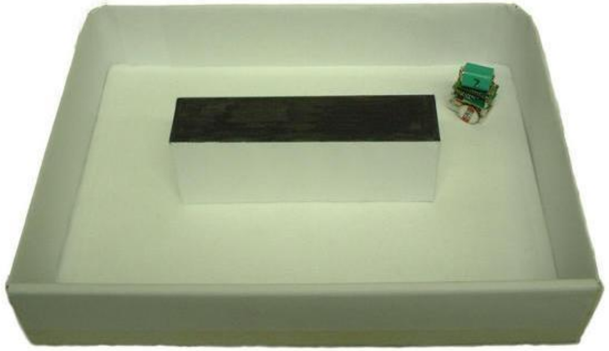
Replace if better than worst in population

?

Mutation

6 individuals
(genome + fitness)

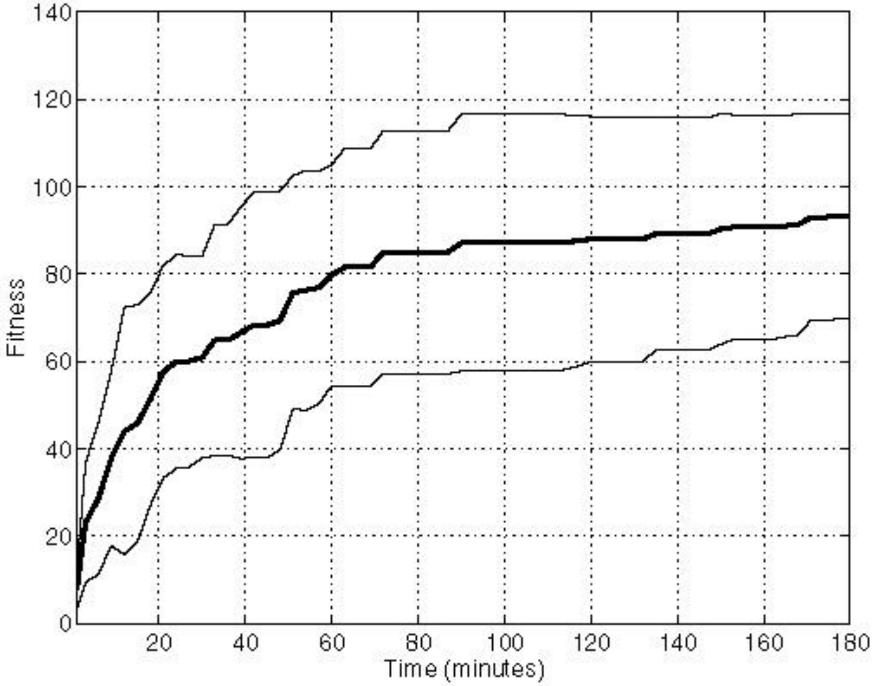
1 bit SIGN
1 bit NCONN
1 bit ICONN

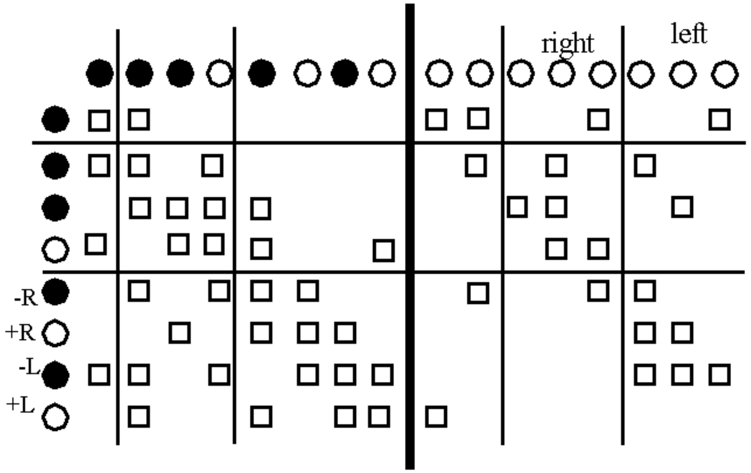


Forward navigation with obstacle avoidance

$$\text{Fitness} = V \times \Delta v \times (1-s)$$

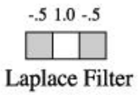
Steady-state evolution

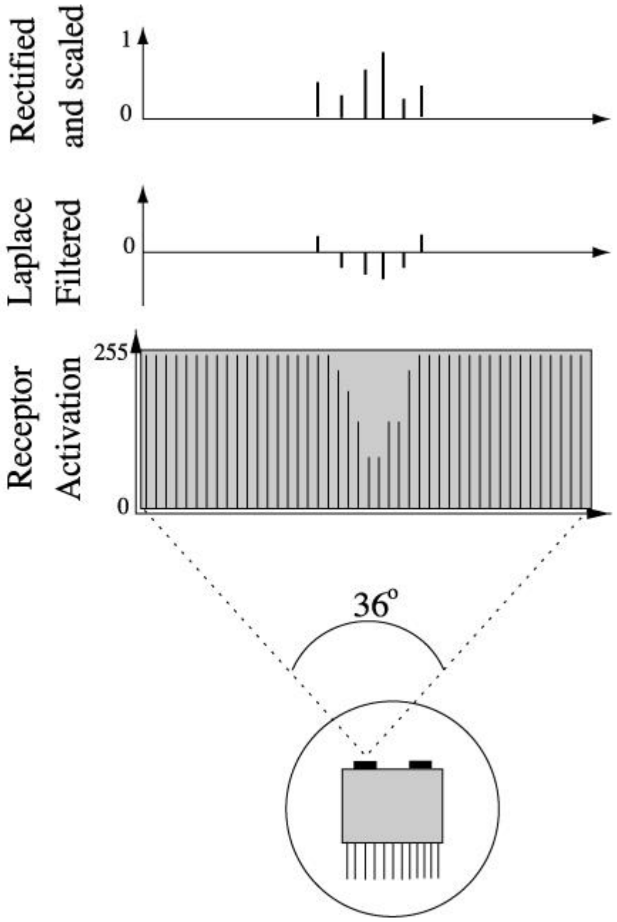




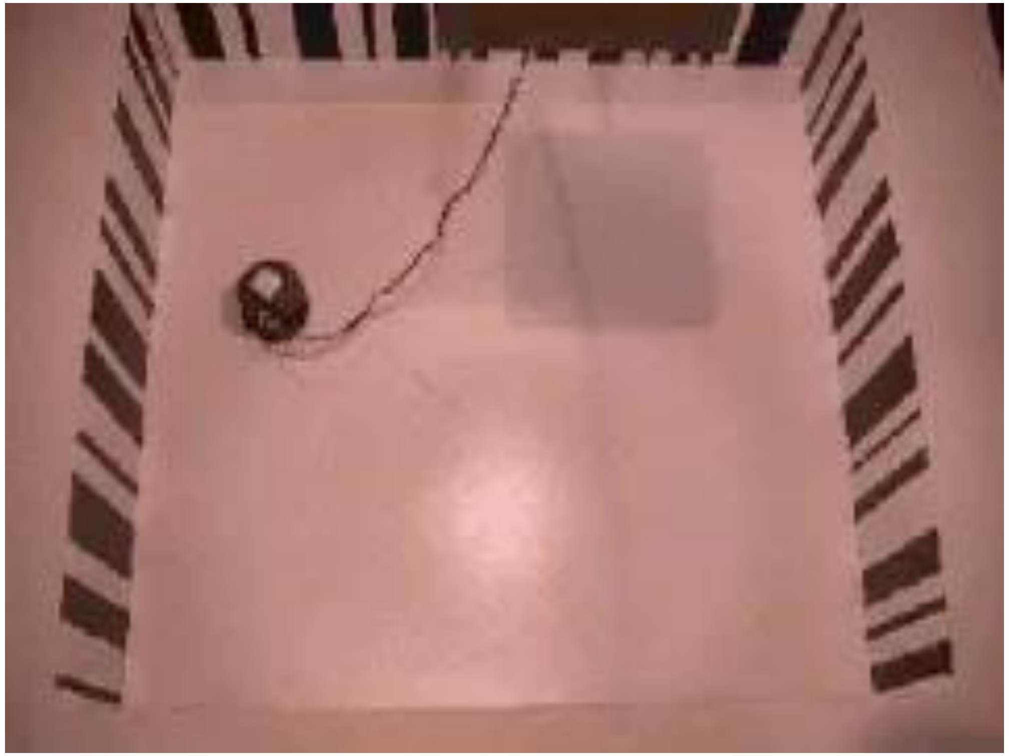
- bias: ↻
- IR Right: ↺
- IR Left: ↻

Vision-based navigation with spiking neurons





Fitness proportional to amount of forward translation over 2 mins

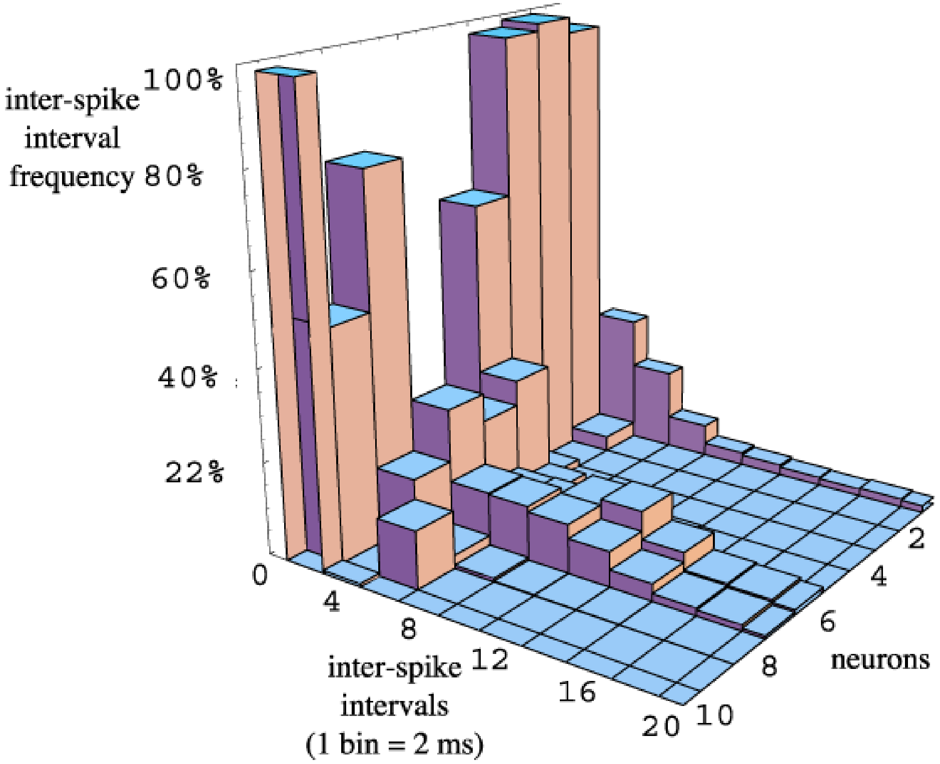


After 30 generations

Firing rate or firing time?

Neuron #

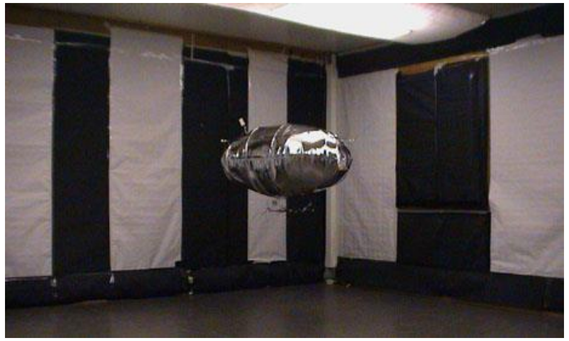
	1	2	3	4	5	6	7	8	9	10
spikes/s	9	445	453	450	330	40	129	363	0	452



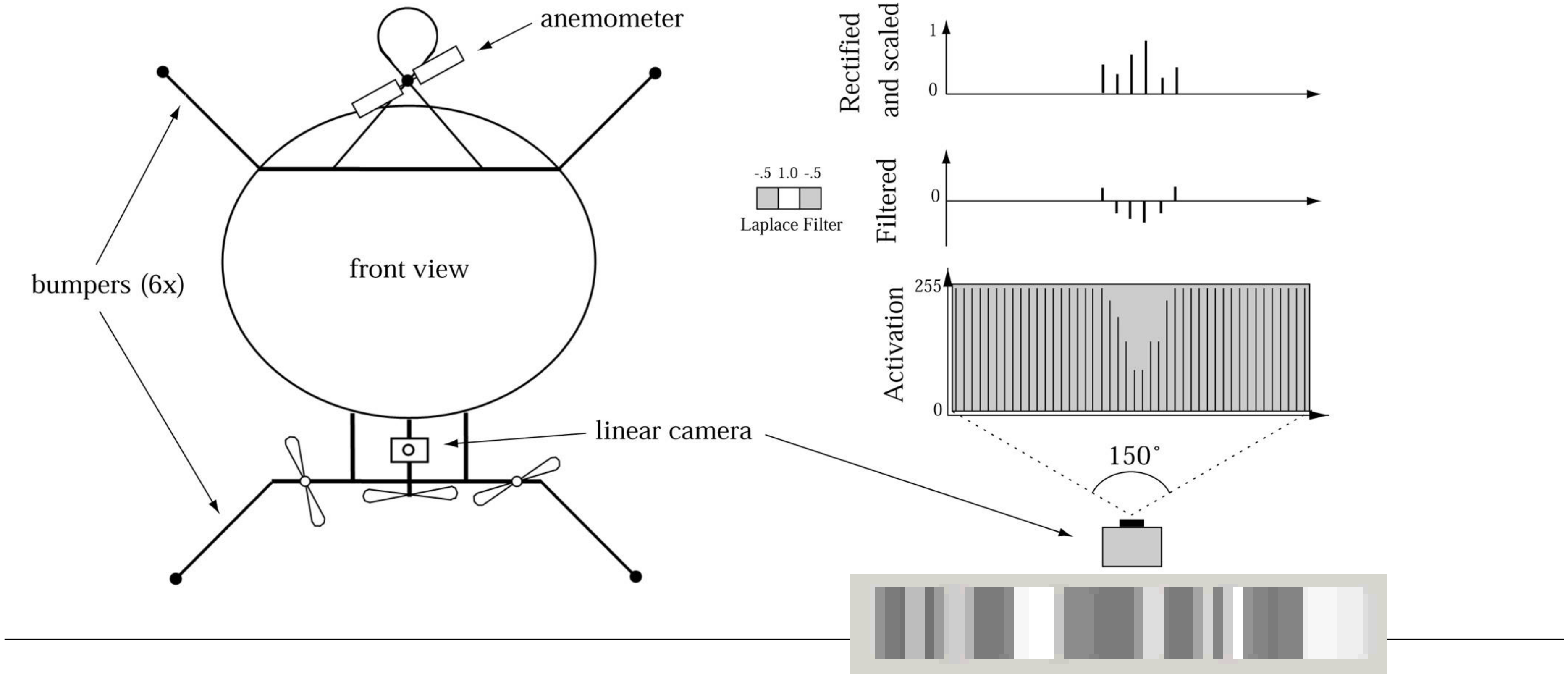
- Removing any single neuron (except # 9) decreases the navigation performance
- Removing any pair of neurons decreases even further navigation performance
- Removing neurons 1, 5, 6 has no effect on performance

we infer that evolved neurons use time difference of incoming signals, not only total signal intensity

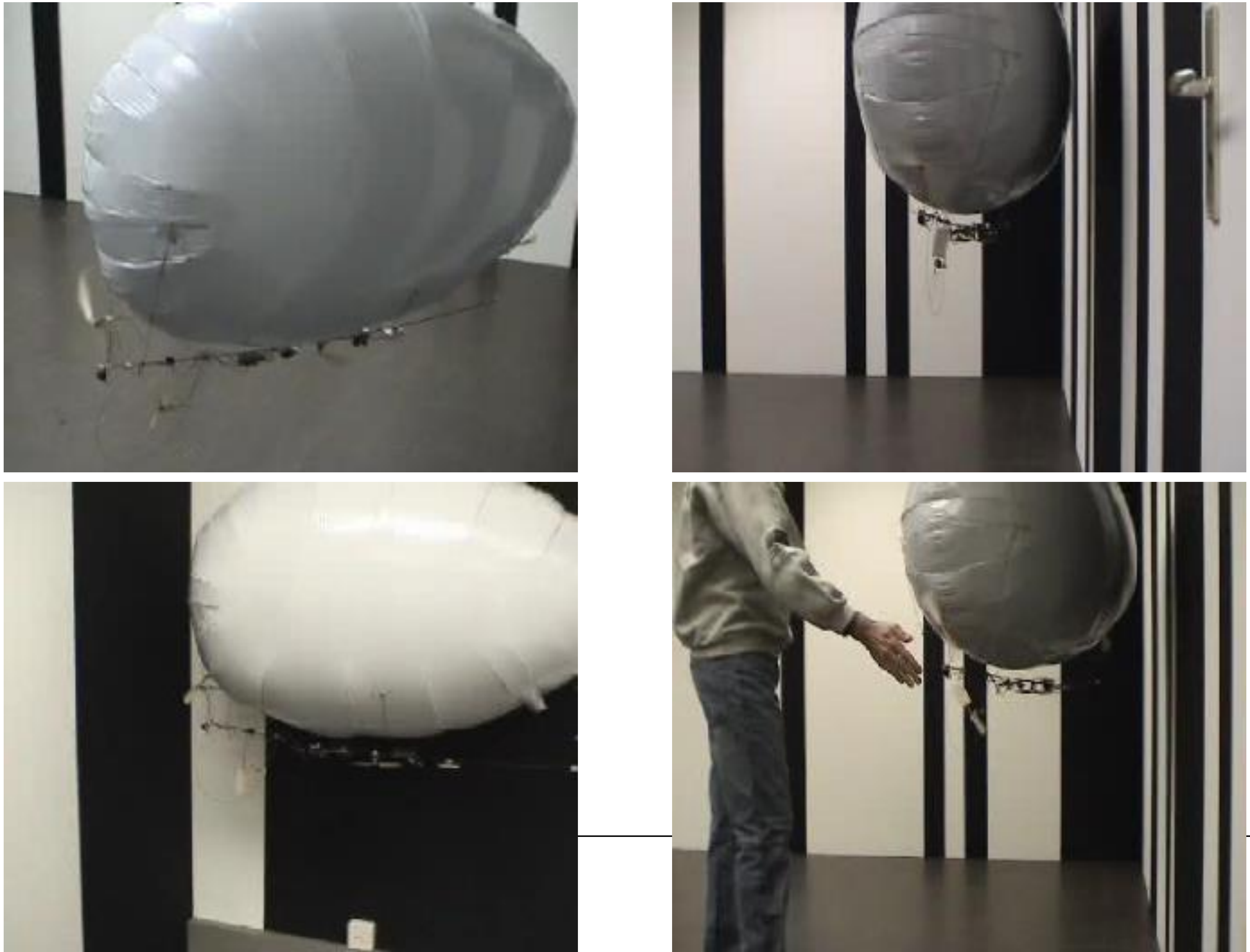
Vision-based flight of a blimp



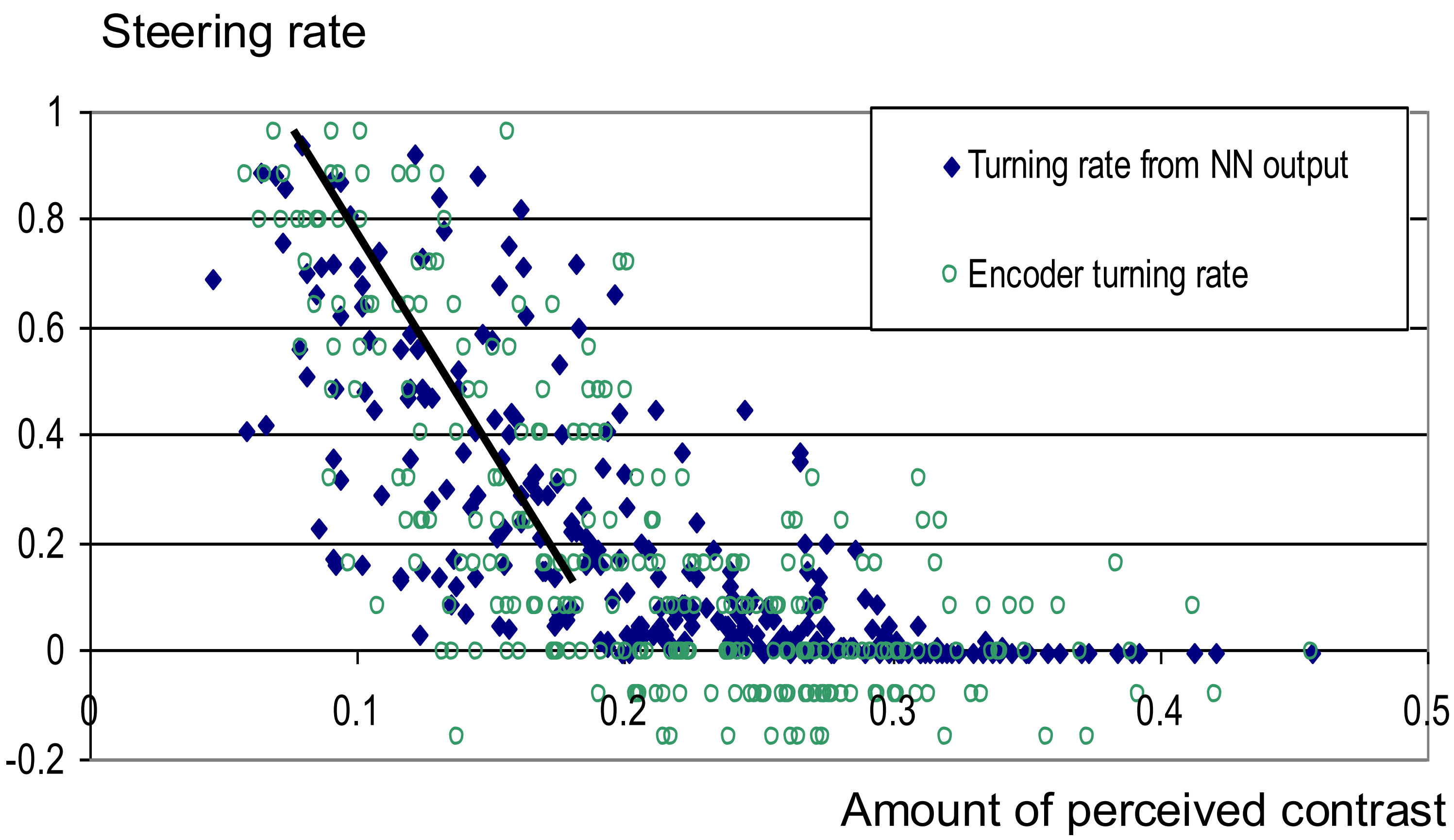
- 5 x 5 room, random size stripes
- Fitness = forward motion (anemometer)
- 2 trials, 2 minutes each
- Evolution + network activation on PC
- Sensory pre-processing on microcontroller



After 50 generations on the real blimp

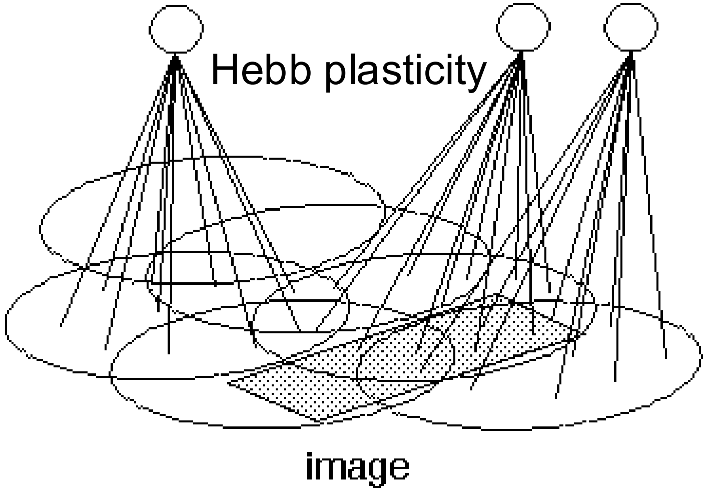


Evolution is opportunistic!

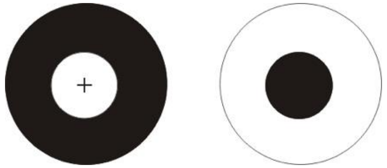


Visual feature detection

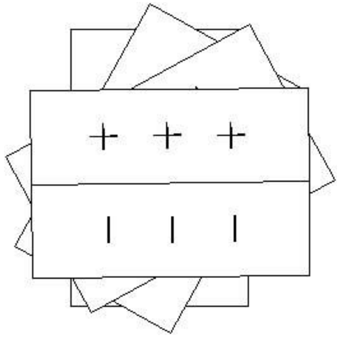
Process whereby visual neurons become sensitive to certain sensory patterns (features) during the developmental process (Hubel & Wiesel, 1959)

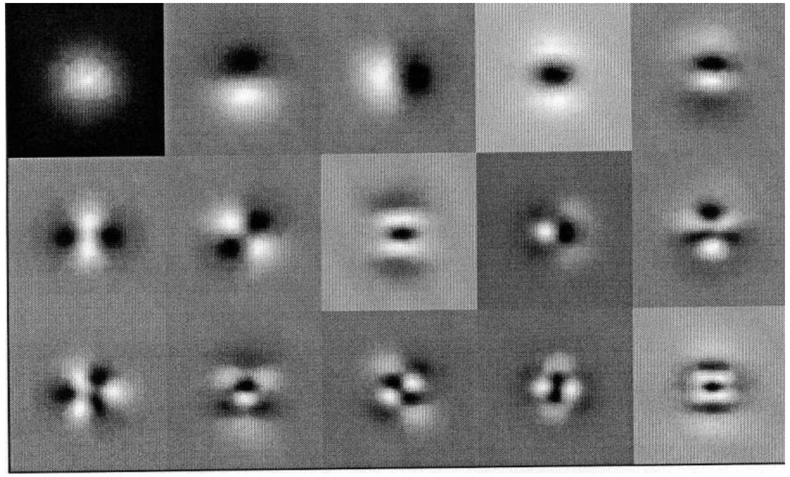


Center-Surround

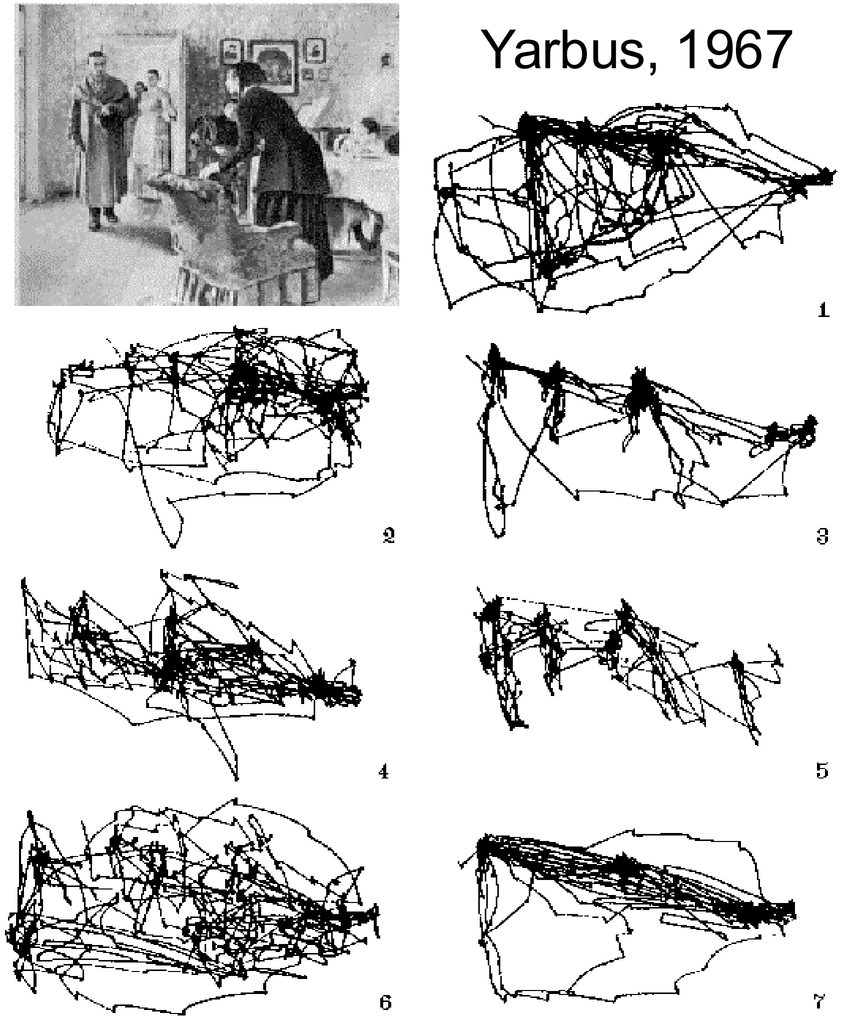


Oriented Edges

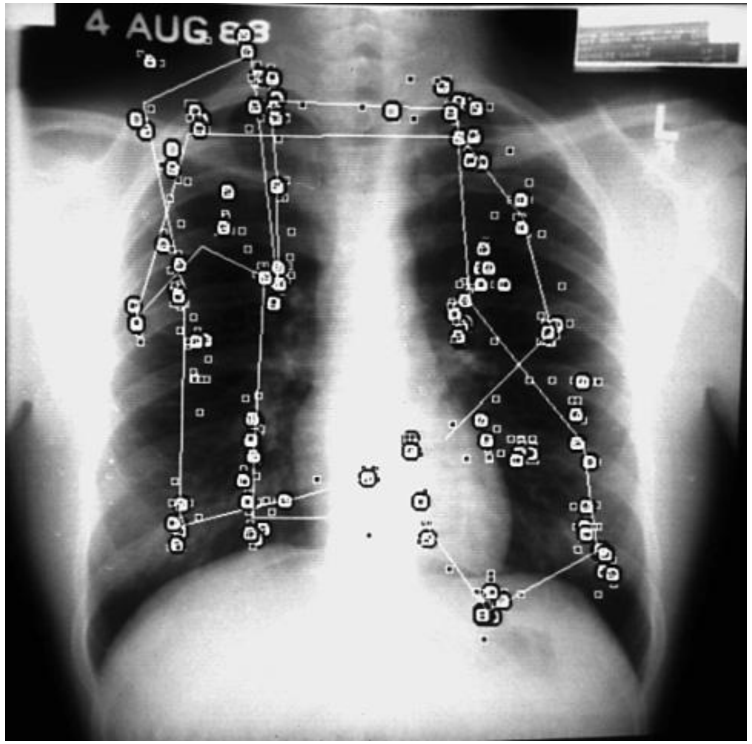




Active vision


Yarbus, 1967

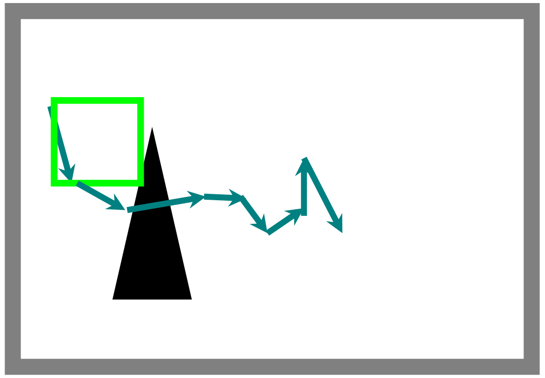
Process of selecting by motor actions sensory patterns (features) that make discrimination easier (Bajcsy, 1988)

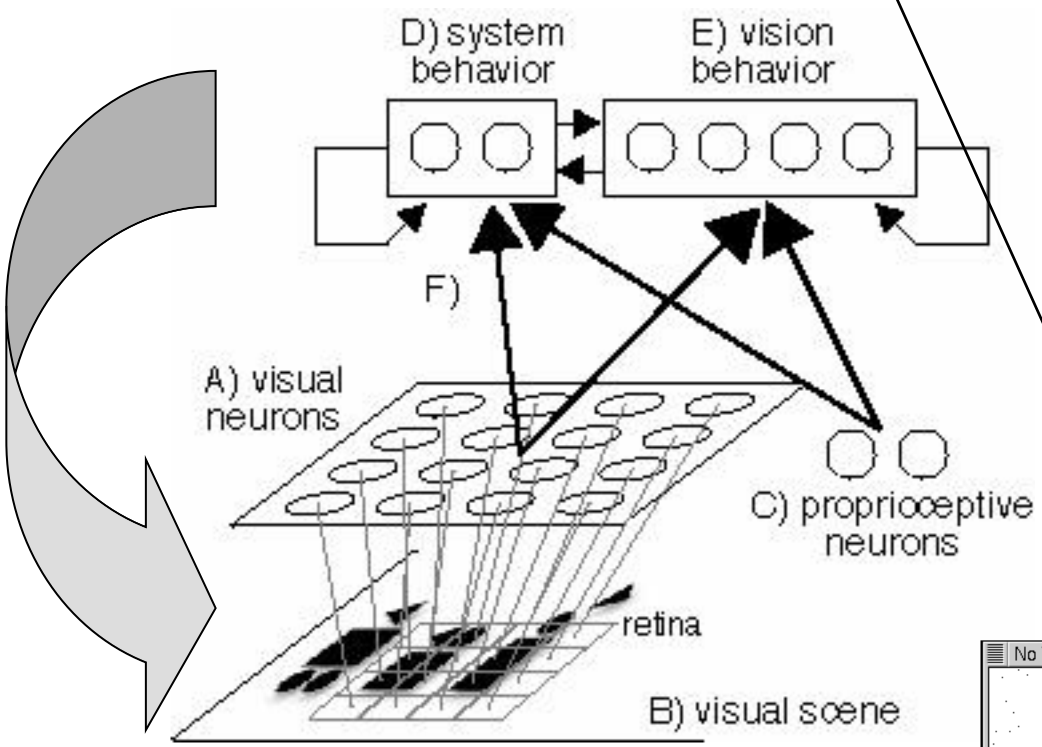


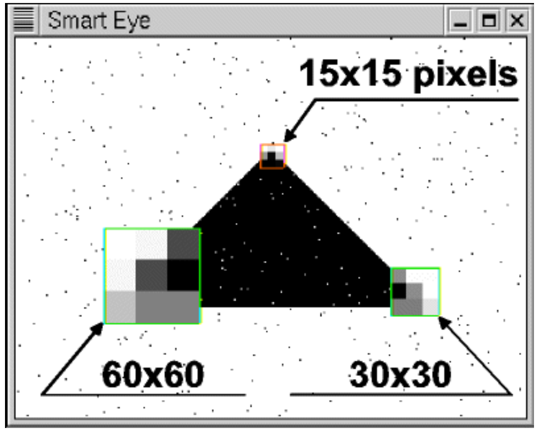
Neural architecture for active vision

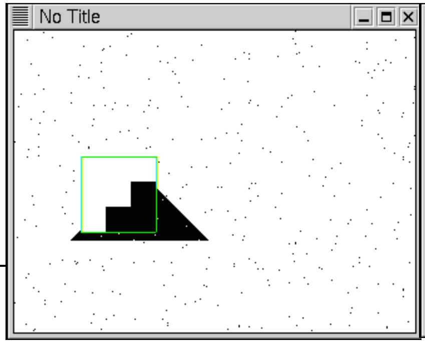
shape discrimination
robot control
car driving

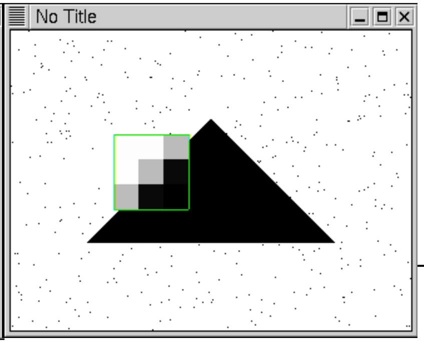
retina movement
zooming factor
filter type

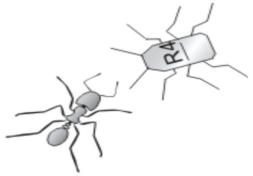








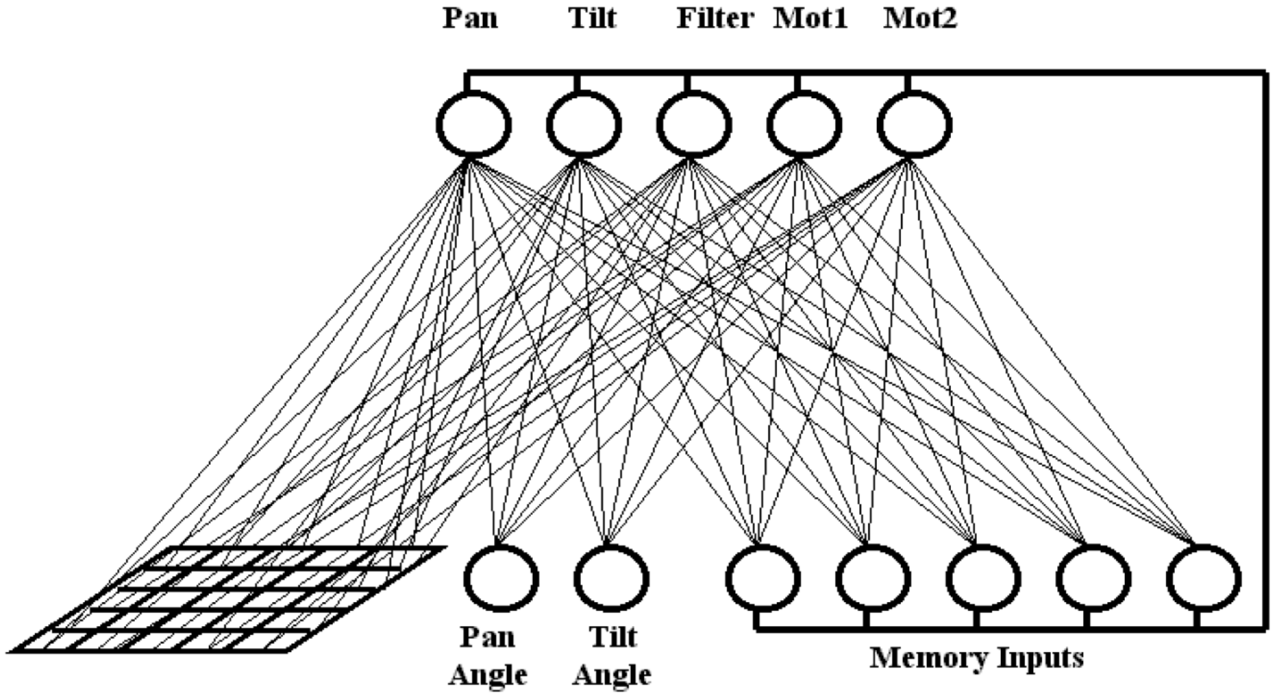




Robot navigation with active vision architecture

Goal: Evolve collision-free navigation using only vision information from a pan/tilt camera.

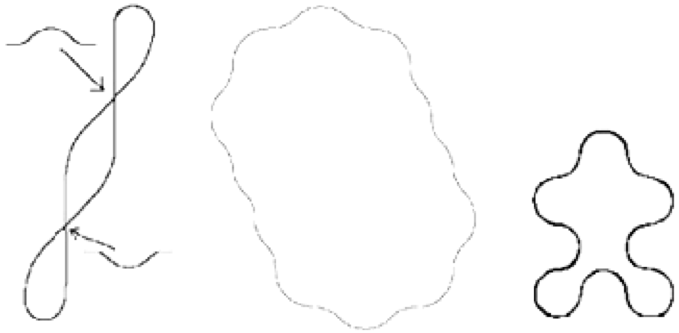
Output of vision system is movement of camera (pan/tilt) and of robot wheels (mot1/mot2). Filter as before.



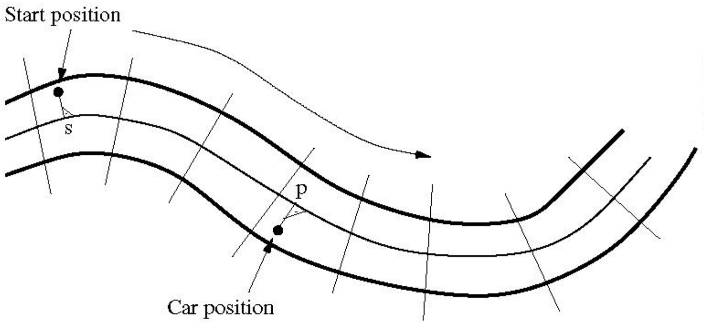
Environment

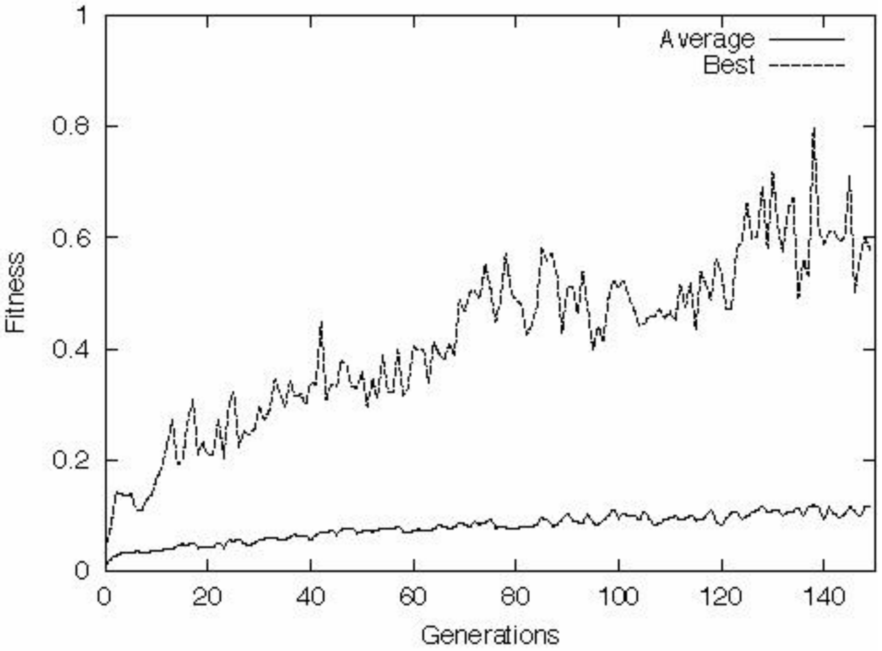
Active Vision for Car Driving

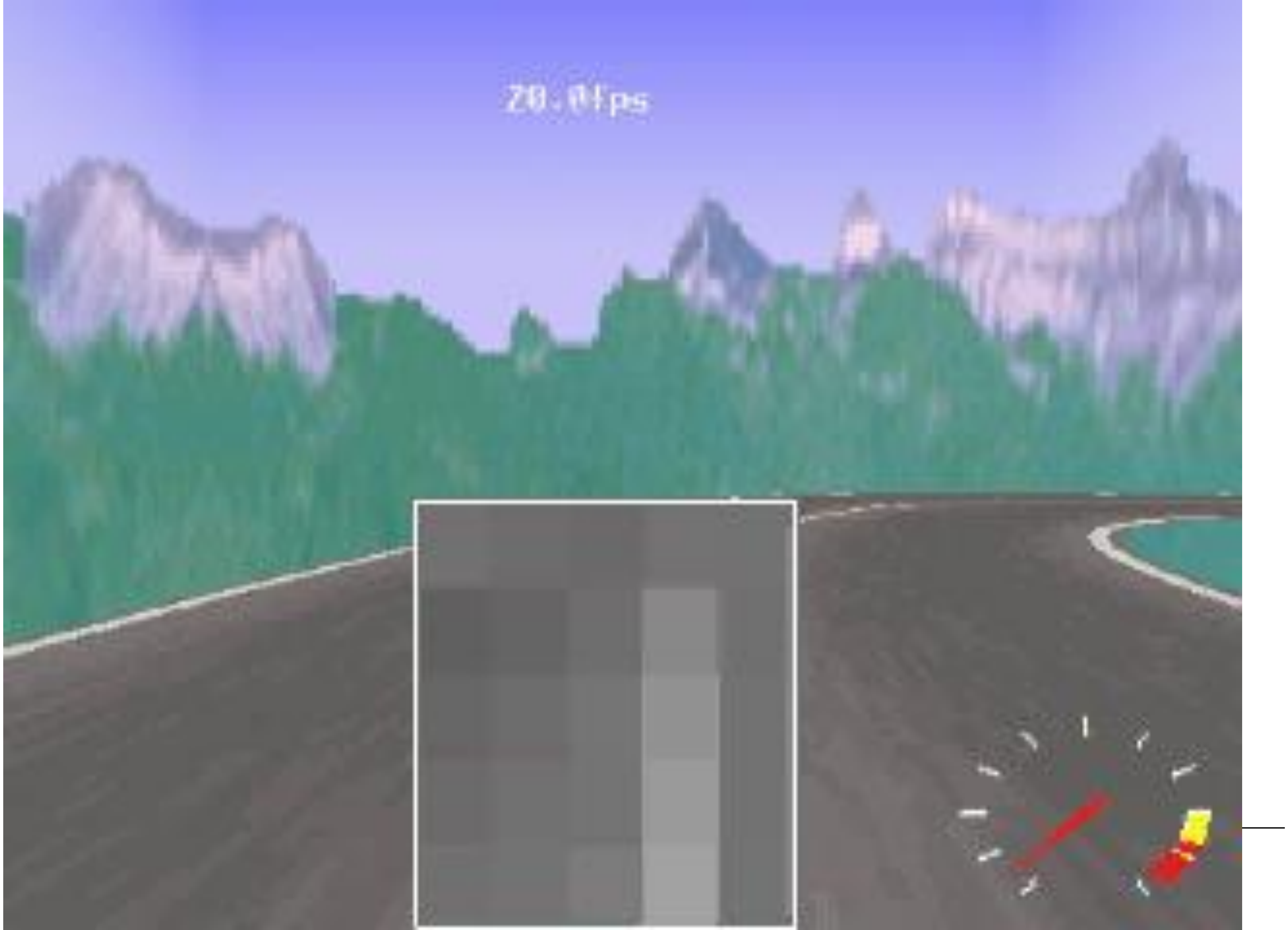
Fitness = percentage of covered distance D in R races on M circuits (limited time for each race).



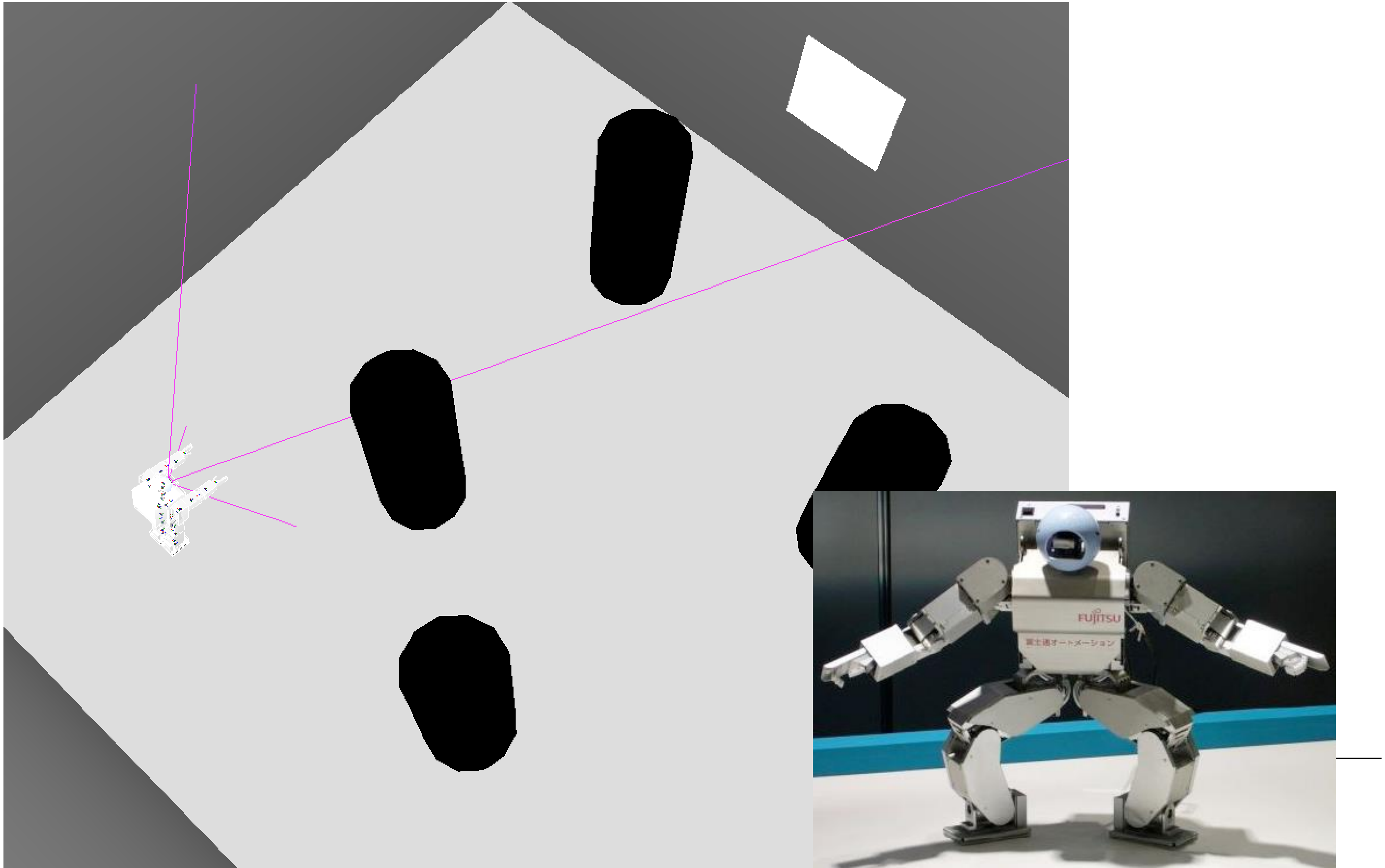
$$F = \frac{1}{R * M} \sum_{r=1}^{R} \sum_{m=1}^{M} D_{r,m}$$

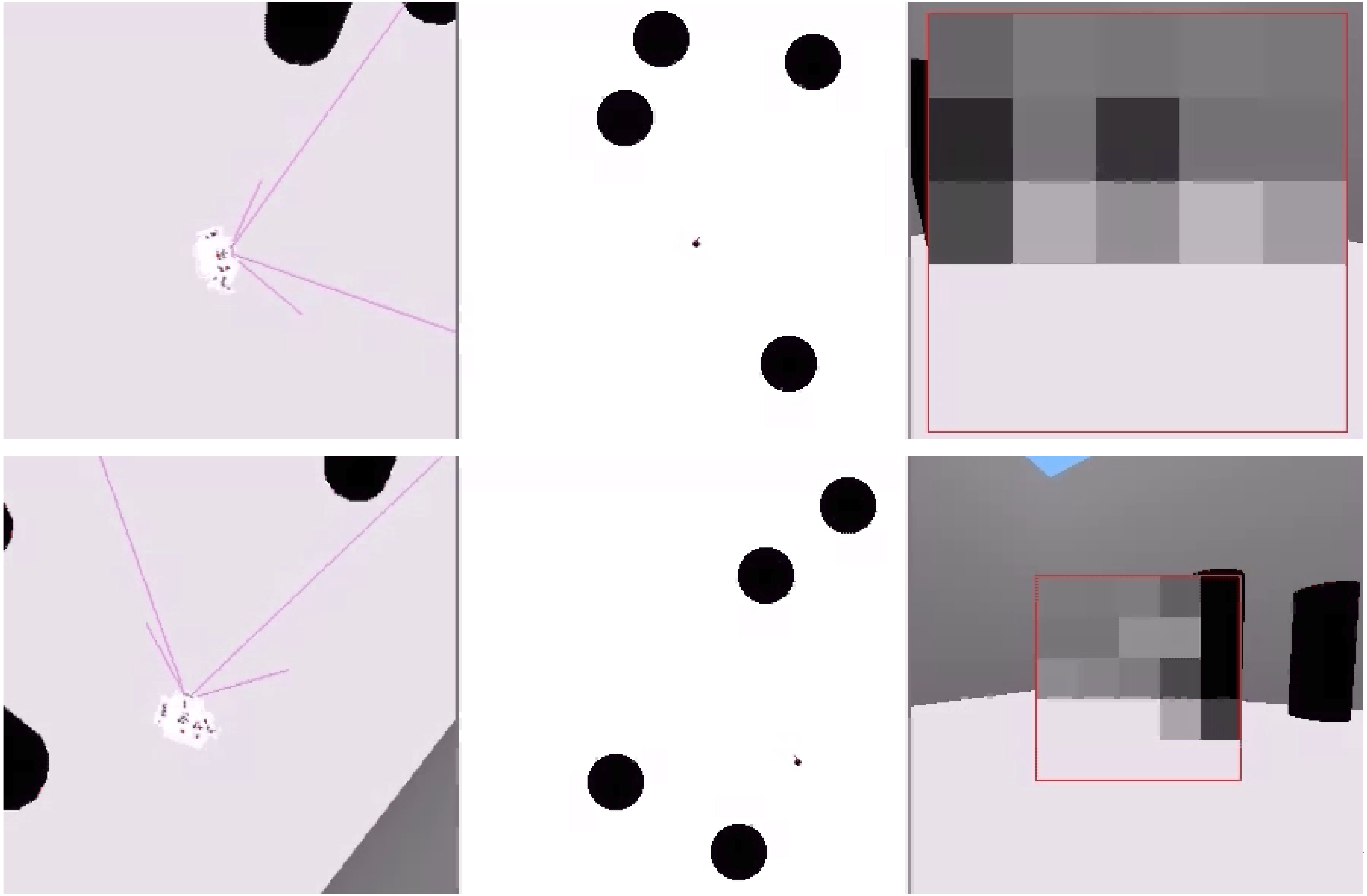




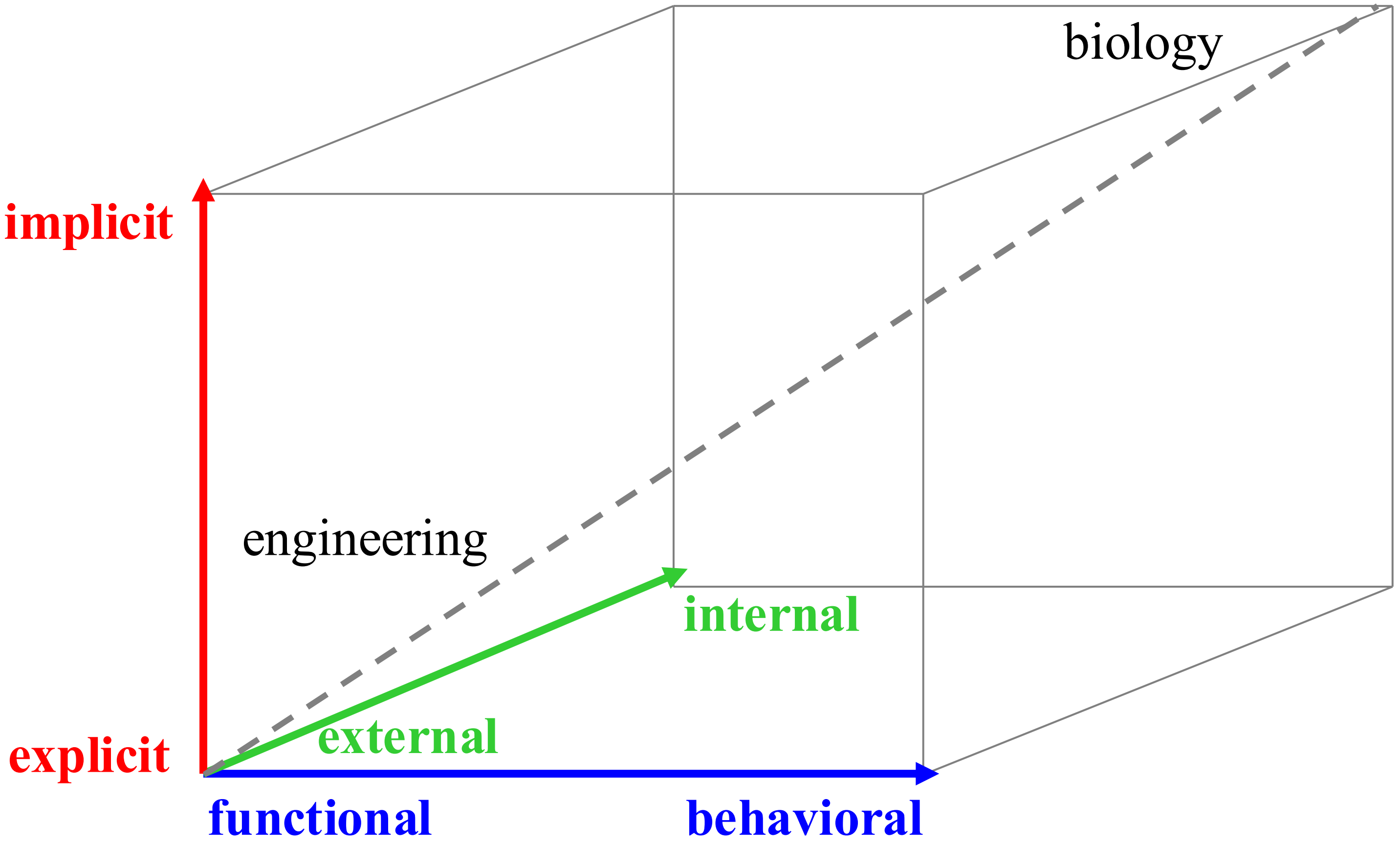

20.0fps

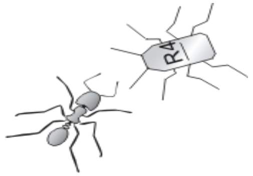
Active Vision for bipedal locomotion





Fitness design space: comparing fitness functions





Checkpoints

- Describe activations of McCulloch-Pitts neurons and of Spiking Neurons
- Describe different types of output functions
- What are neural receptive fields
- What is a bias unit and how is it included in the network computation
- Describe types of neural architectures
- Methods for encoding sensory signals
- Genetic encodings of neural networks
- Give a robot configuration and a navigation task, choose the neural architecture, state and action encoding, evolutionary algorithm, and fitness function
- Describe the Steady-State Evolutionary Algorithm