NEURAL NETWORKS AND GENETIC ALGORITHMS

Buzzoni Marco (marco.buzzoni2@studio.unibo.it)
Francesconi Alessandro (alessand.francescon2@studio.unibo.it)

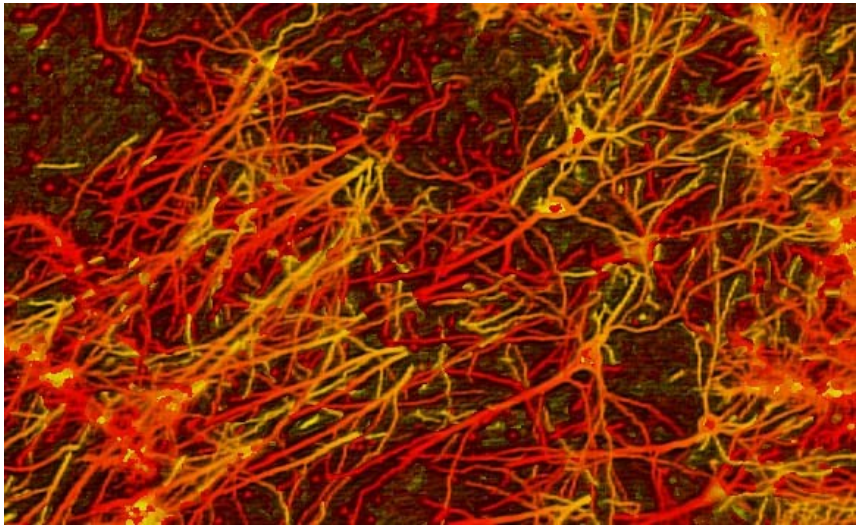
Università degli Studi di Bologna
LM Informatica 2012/13

Today's outline

- Neural Networks
  - The Neuroscience: introduction to biological neural networks
  - From biological to artificial neural networks
  - Different kinds of networks
- Genetic Algorithms
  - A "natural" selection
  - Methods for creating a new generation
- A little project…

The Neuroscience

- Our brain is, substantially, a Parallel Information Processing System.
- It contains about 10 BILLION nerve cells, called **neurons**, each one connected to other ones by **synapses**.



The Neuroscience

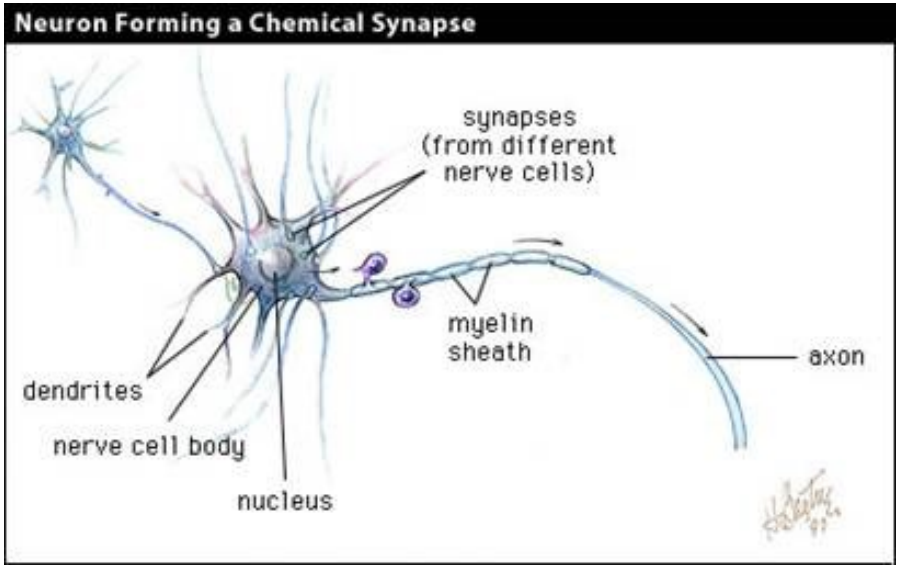
- A **biological neural network** is an interconnection of neurons whose sequential or parallel activation is defined with a precise logic.

BUT WHAT IS AN

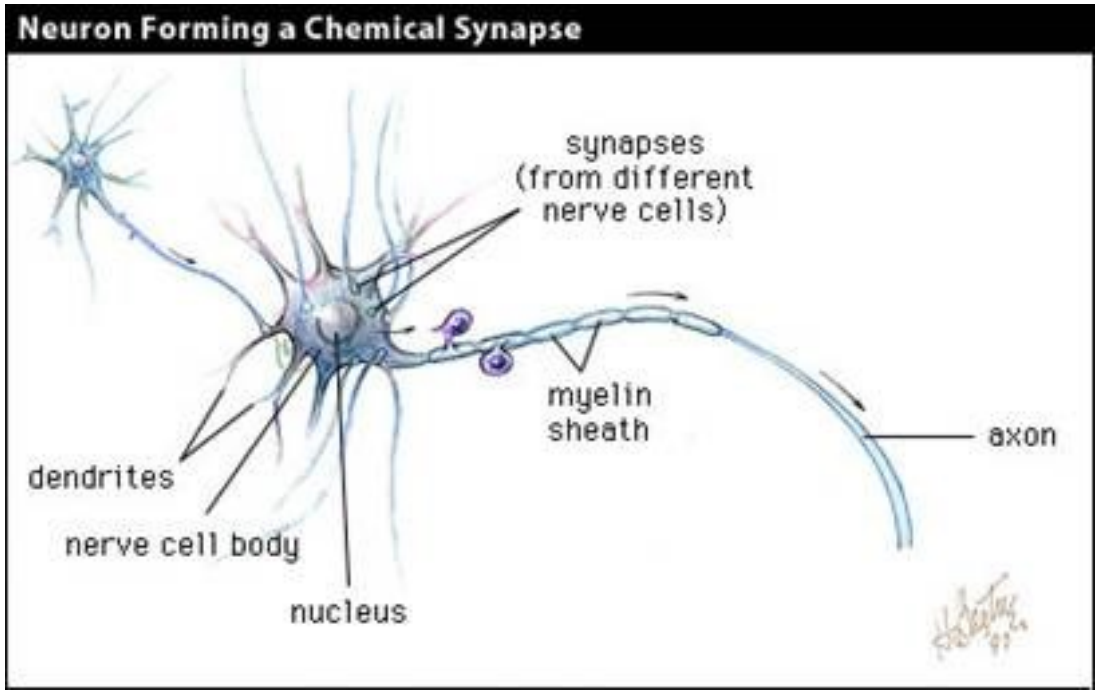
"ACTIVATION"?

The Neuroscience

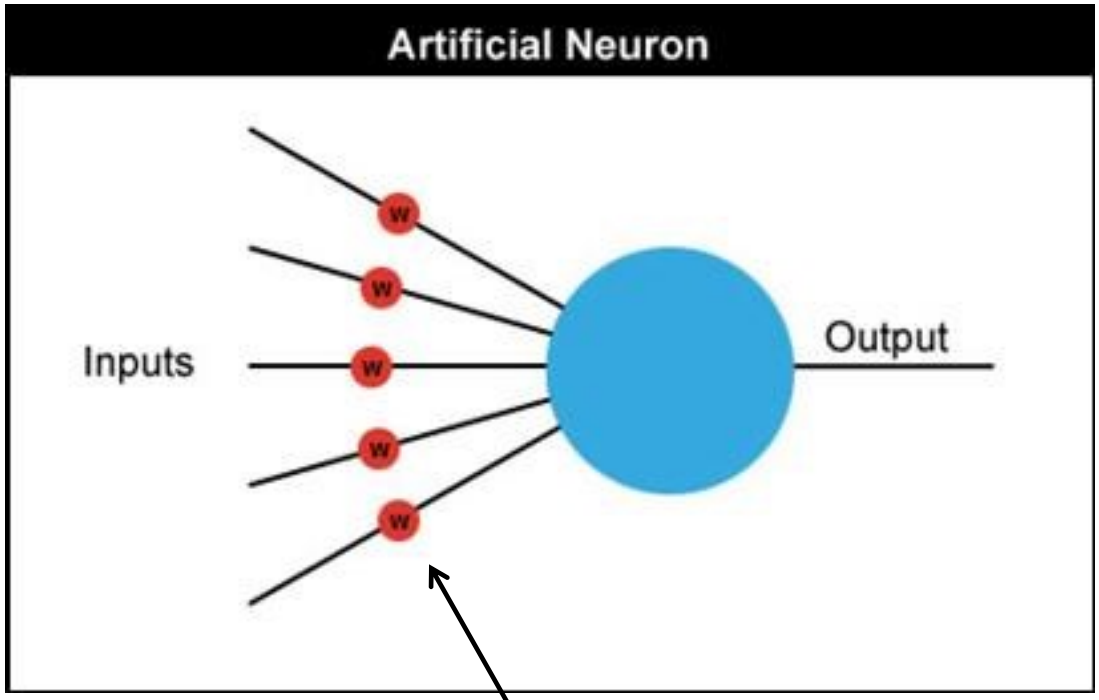
- The neuron continuously receives signals from these inputs and transmits informations using a little bit of… **magic**.
- For our purposes, we can simplify this action in a way such that a neuron outputs **a value that is strongly influenced** by those inputs that have a higher "importance factor", called *weight.*



Biological neural networks



Artificial neural networks



"Weights" are simply signed floating point numbers.
Initial value is random.

Artificial neural networks

- The activation is made by checking this equation:

$$x_1w_1 + x_2w_2 + x_3w_3 \ldots + x_nw_n \geq t$$

  - x_i is the i^{th} input for $1 \leq i \leq n$
  - w_i is the weight of the i^{th} input
  - t is a threshold value

- So, the activation depends upon whether or not the left sum exceeds the threshold value. How big is it? We can do this way…

$$x_1w_1 + x_2w_2 + x_3w_3 \ldots + x_nw_n - t \geq 0$$

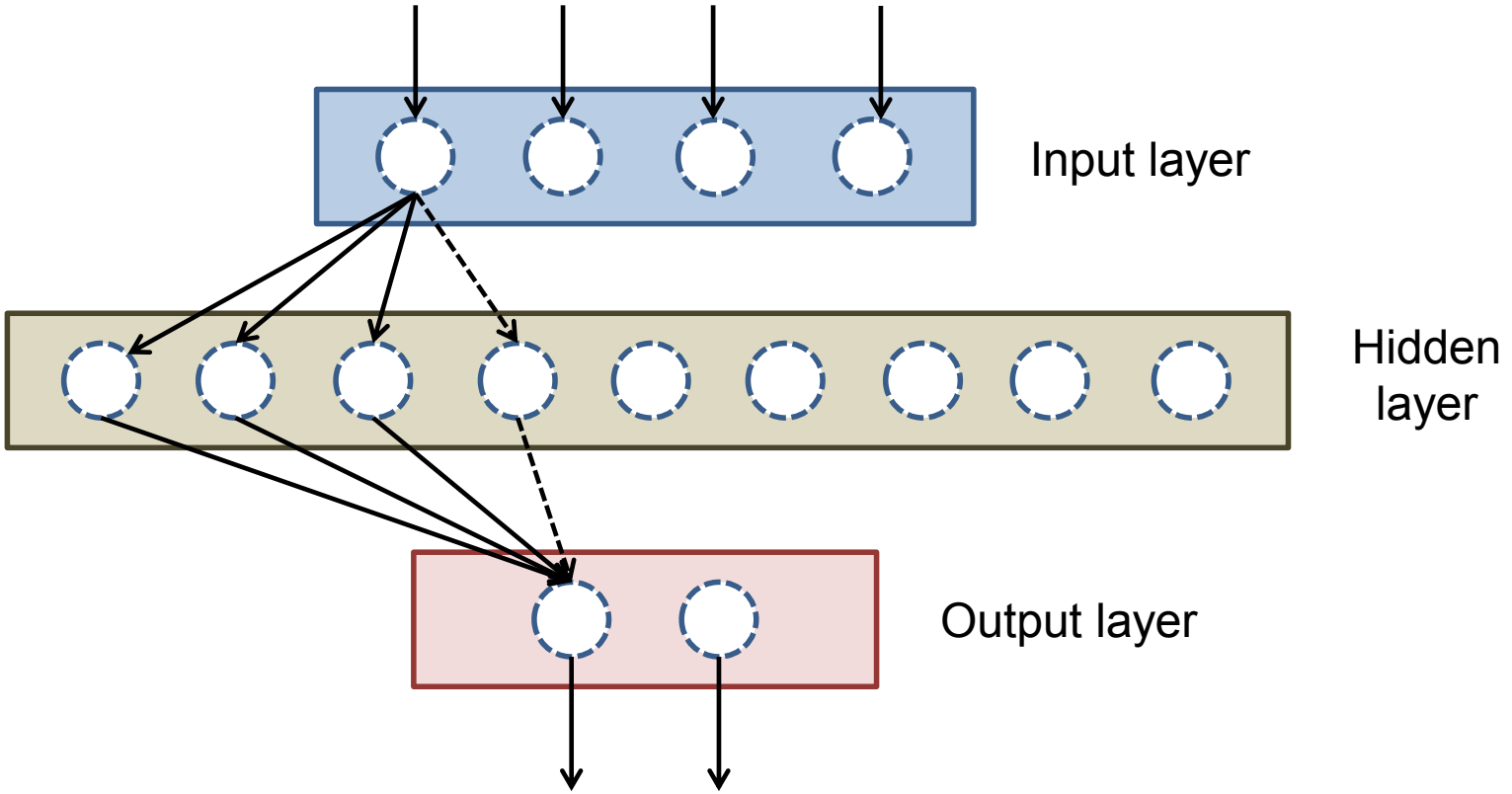
- That is equal to

$$x_1w_1 + x_2w_2 + x_3w_3 \ldots + x_nw_n + \mathbf{(-1)t} \geq 0$$

- Threshold is now a further weight multiplied by a constant called **bias**

We made a neuron! … but how to build the **network**?

- For most of the problems, an artificial neural network is a data structure composed by 3 layers:



Artificial neural network

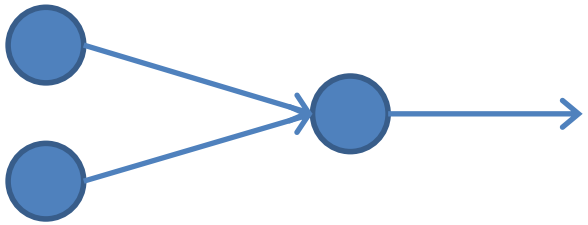
- *Input layer* holds **the input data coming from the problem**, for example, the current state of an environment or the input obtained from the agent's sensors, properly discretized.

- *Output layer* contains **the solution to the problem**, that can be more or less complex:
  - From one resulting number (there will be only one neuron)
  - To a set of *n* numbers ("do action X with properties Y, W, and Z")

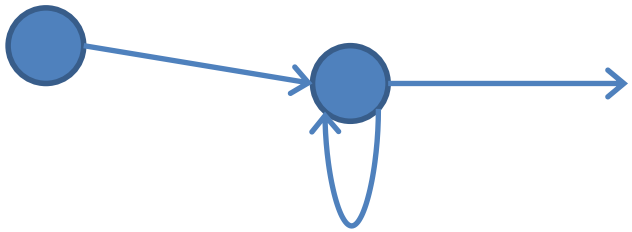
- How about the *Hidden layer*? Works as a **mapper** between inputs and outputs. The number of internal neurons specifies the precision of the outputted data.

Different kinds of neural networks

- Based on the data flow:
  - Feed-forward: there is no need to synchronize the inputs of a neuron.



  - Recurrent: there is loops inside the neural network.



Different kinds of neural networks

- Based on learning algorithm:
  - Supervised learning: the expected output is known, the neural network is trained to map the input values to the right output.
  - Unsupervised learning: the expected output is unknown, the task of the neural network is to **adjust the weights itself** to obtain a correct output.

Weights

- *Weights* are the core elements of a traditional neural network. Setting weights in the right way will led to a good (*expected*) result set.

- But our goal is to **automate the process** of weights calculation trought a method based on the just passed experience!

WHO DOES THE

DIRTY JOB?

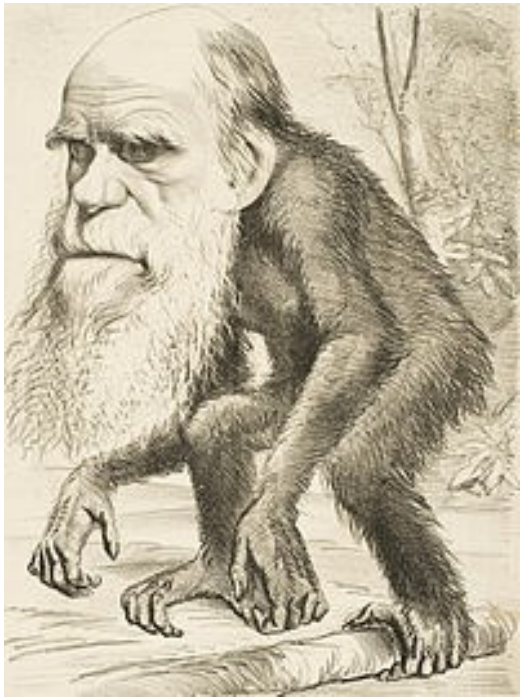
Genetic algorithms

- Genetic algorithms are a family of search heuristics that **mimics the processes of natural evolution**, such as *inheritance*, *mutation*, *selection*, and *crossover*.

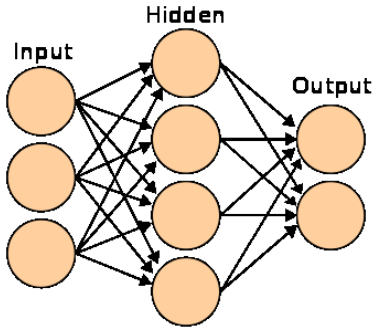
- With such methods, we can literally give birth to and raise a **population of chromosomes**, each one containing a candidate solution to the problem.

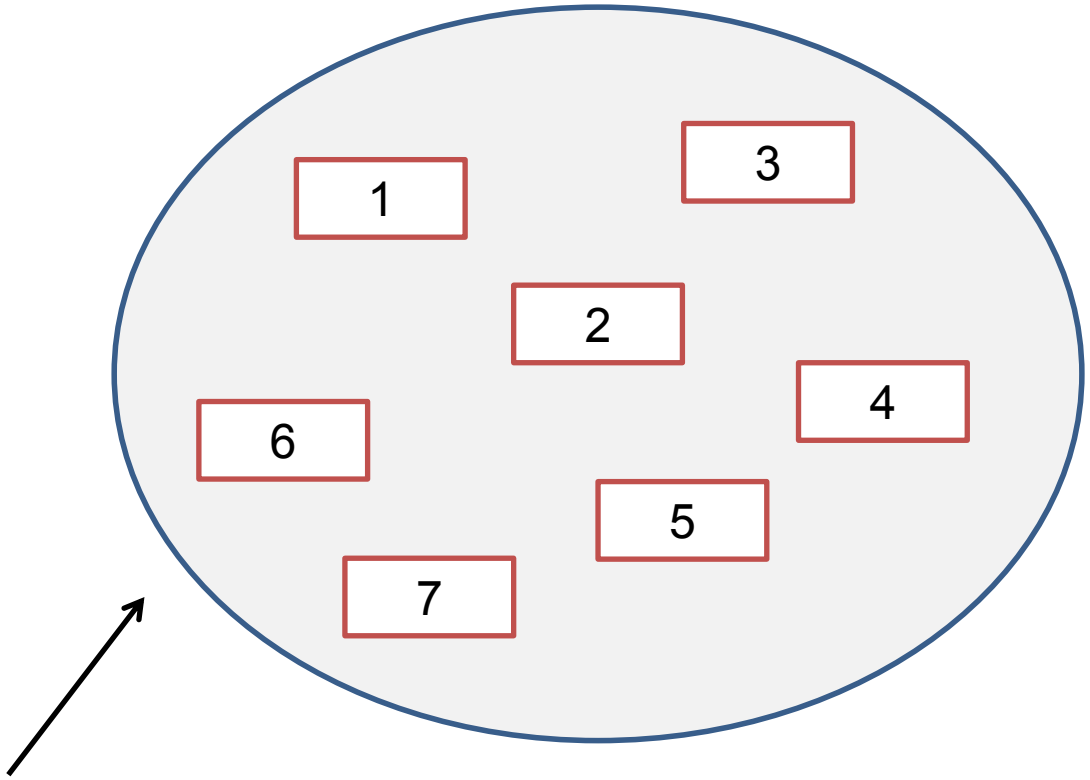
- In our case, each chromosome will contain a possible weight set for the neural network.

- … the **best** will ***survive***…



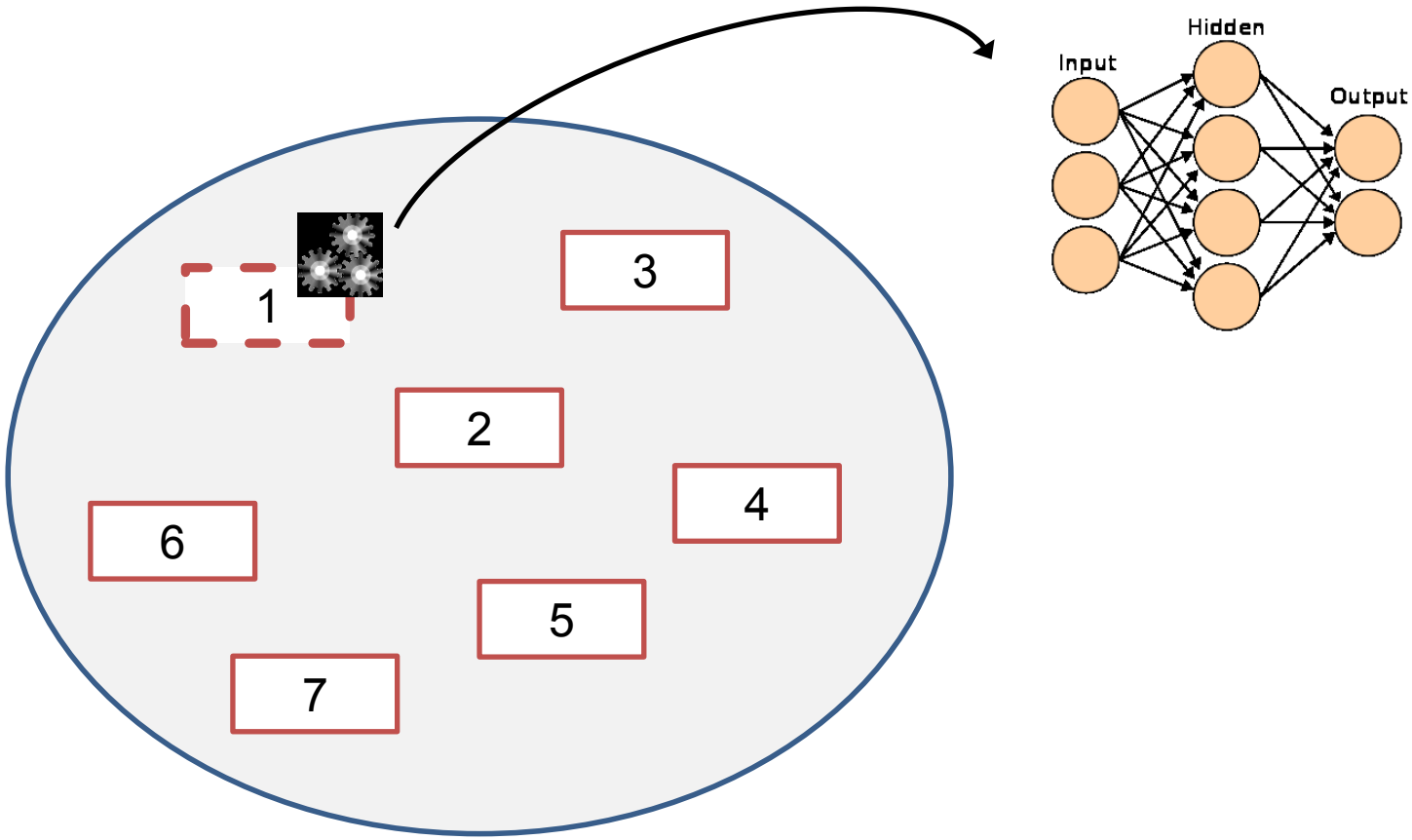
Genetic algorithms





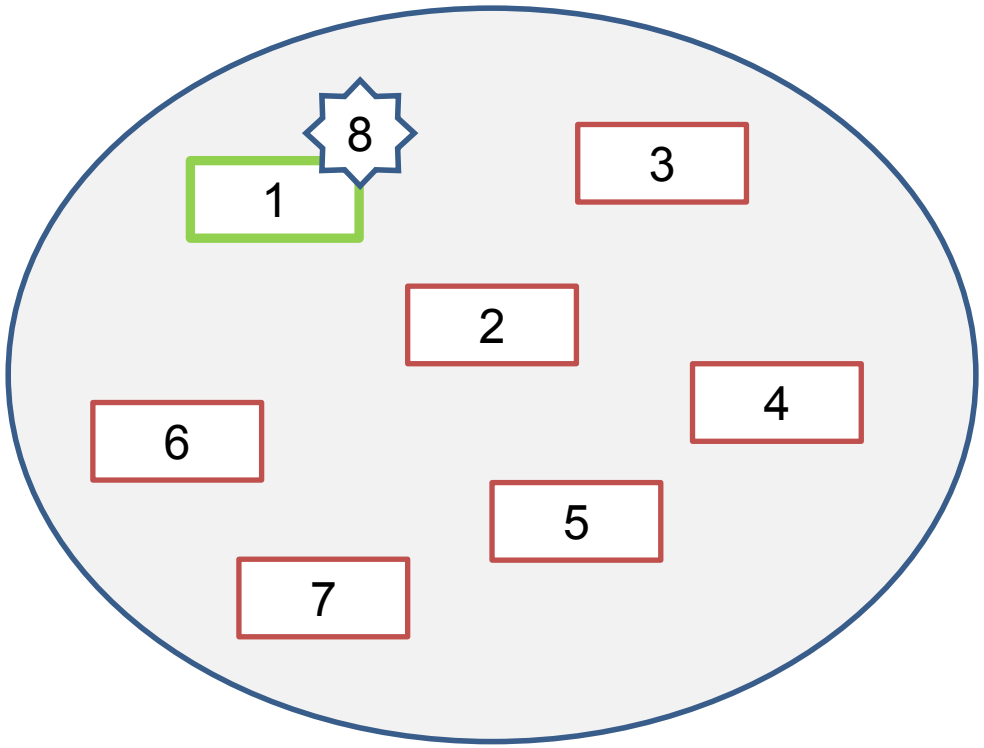
This is a **generation** of chromosomes

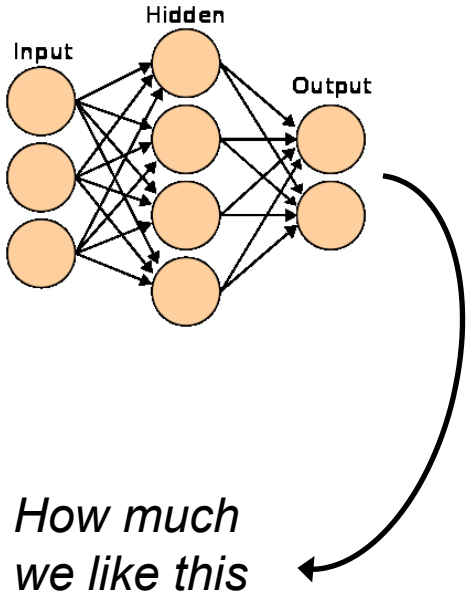
Genetic algorithms



We test each chromosome's effectiveness and give a **score** to it

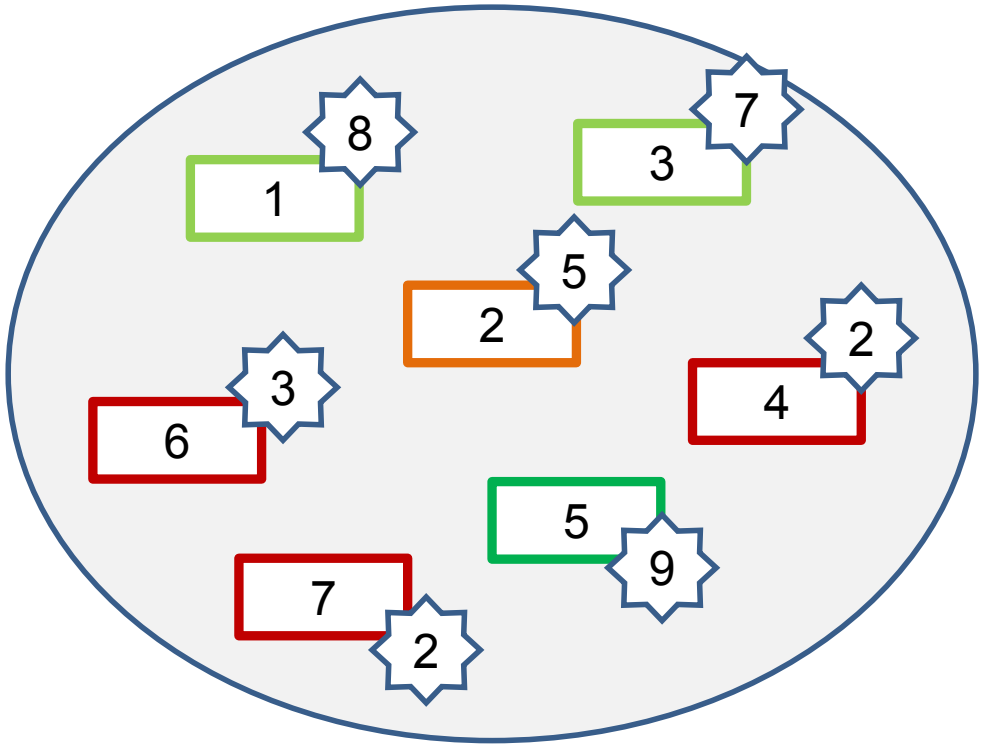
Genetic algorithms

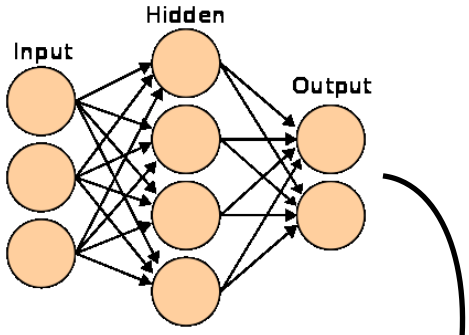




We test each chromosome's effectiveness and give a **score** to it

Genetic algorithms

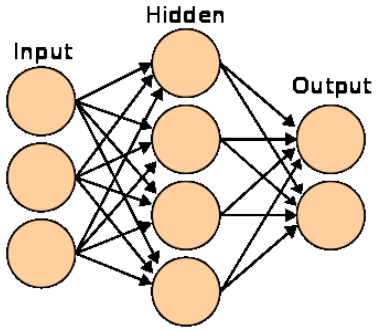


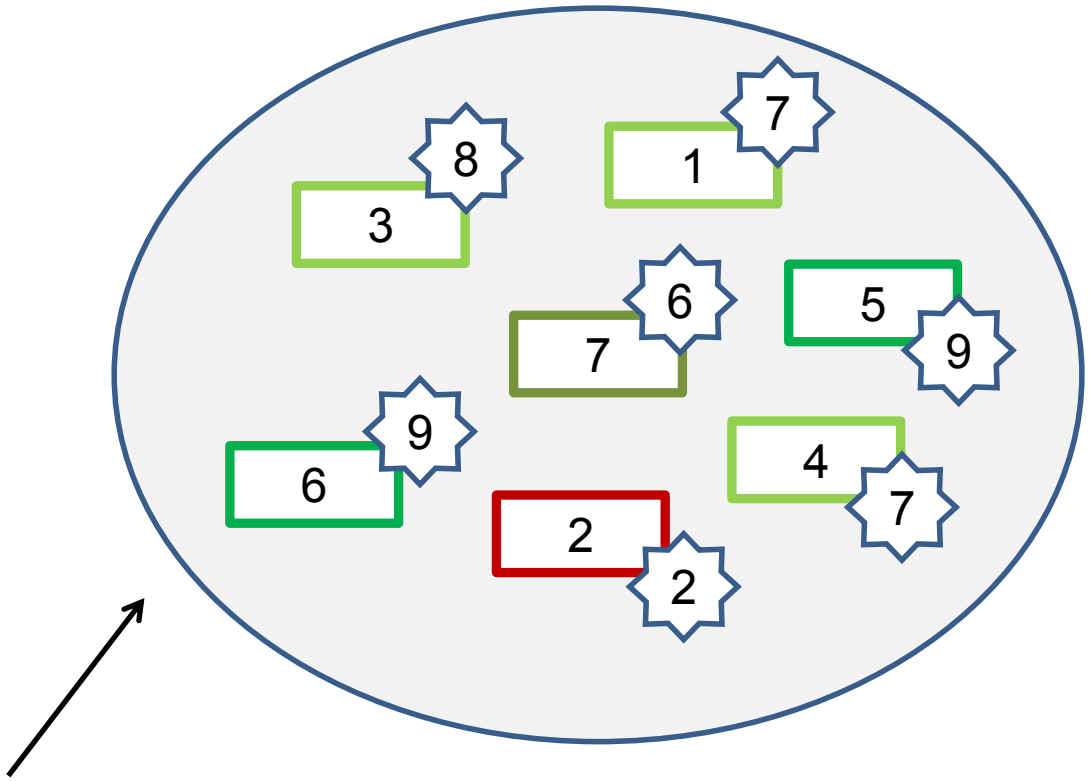


"How much do we like this output?"

We test each chromosome's effectiveness and give a **score** to it

Genetic algorithms





The **next generation** will contain better chromosomes!

The fitness value

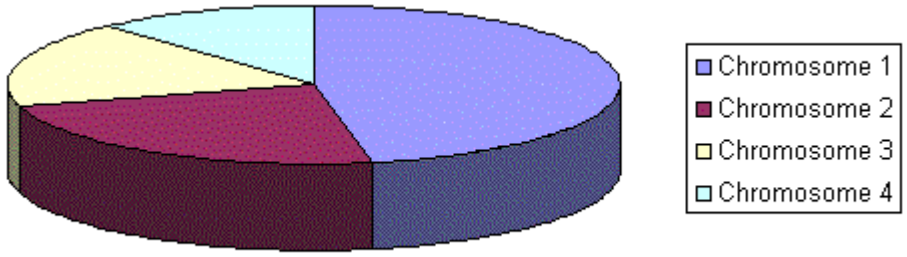
- We talked about a score gived to each chromosome. A better known name for this number is *fitness*.

- In each generation, the fitness of every individual is evaluated. Then, **multiple individuals are selected, recombined and mutated** to form **a new population**.

- The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a **satisfactory fitness level has been reached.**

Selection

- The practice of select **the best candidates** can be done in various ways. A widely used method is called "Roulette Wheel".

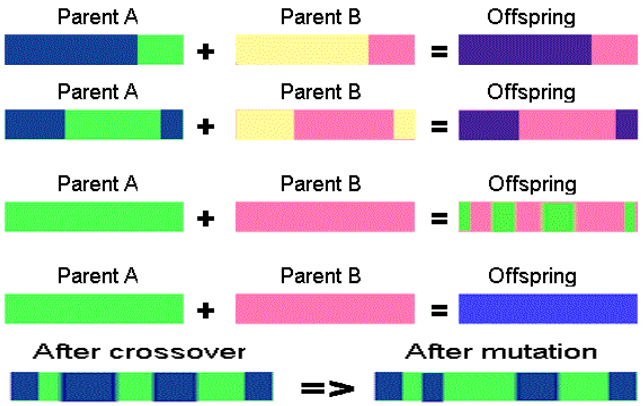



```
1. Calculate sum S of all chromosome fitnesses in population
2. Generate random number R from 0 to S
3. While S' < R
        Go through the population and sum fitnesses in S'
4. Return the chromosome where you are
```

Recombination

- By recombining two chromosomes it is possible to obtain **a couple of better chromosomes** (*offspring*). This operation is also called *crossover*.

- Different kinds of crossover:
  - One point
  - Two point
  - Uniform
  - Arithmetic
  - Bit inversion



- It's suggested to set the probability to have a crossover for each new generation between 80% and 95%

Mutation

- Mutation is used **to avoid local minimum** of the fitness function. It is performed by changing elements of a chromosome of a small amount.

(1.29 5.68 **2.86** **4.11** 5.55) => (1.29 5.68 **2.73** **4.22** 5.55)

- It is suggested to perform a mutation before any new generation with probability between 0,5% and 1,0%.

References

- For Neural Networks:
  - R. Rojas: *Neural Networks – A systematic introduction*
    - Springer-Verlag, Berlin, New-York, 1996.
    - Free online PDF: http://page.mi.fu-berlin.de/rojas/neural/index.html.html
  - AI Junkie - Neural Networks in plain english
    - http://www.ai-junkie.com/ann/evolved/nnt1.html
- For genetic algorithms:
  - Introduction to Genetic Algorithms
    - http://www.obitko.com/tutorials/genetic-algorithms/index.php
  - D. Whitley: Genetic Algorithms and Neural Networks (1995)

CARWIN

A self-driving car implemented with a neural network and a genetic algorithm (AI Course Project)

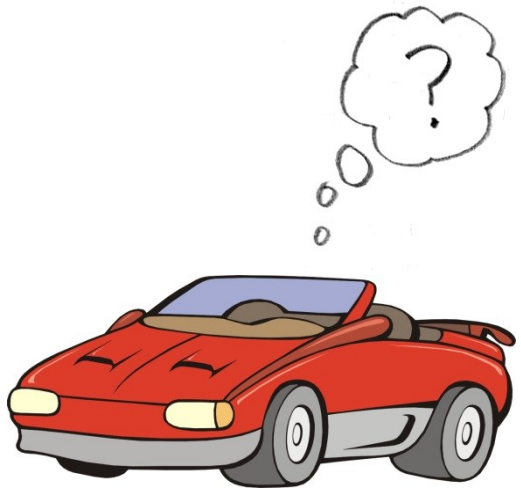
Università degli Studi di Bologna
LM Informatica 2012/13

Carwin

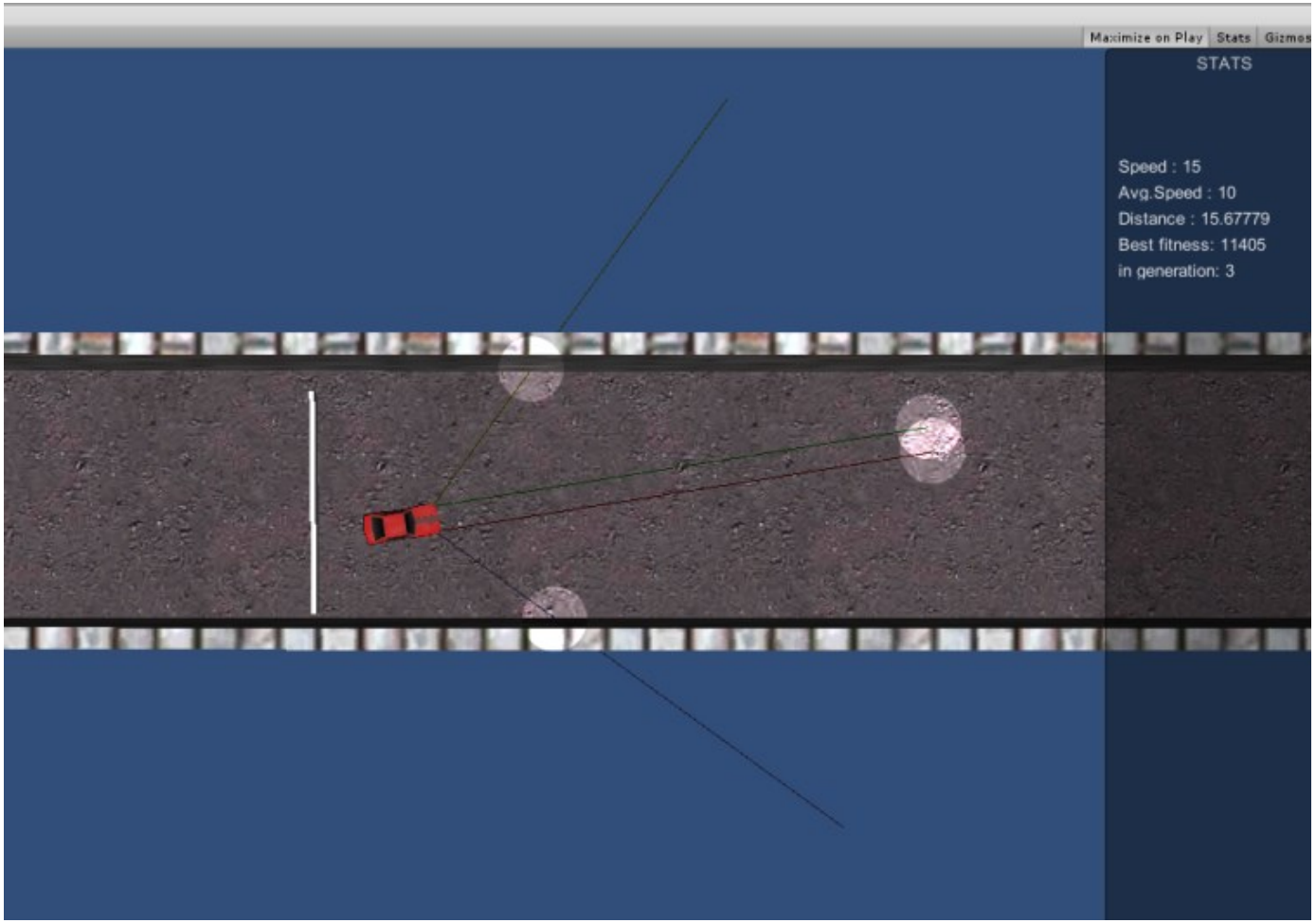
- Carwin is a car driven by a neural network which is trained by a genetic algorithm.

- The neural network inputs are:
  - Values returned by 4 sensors using raycast method:
    - Angle of a imminent turn, in degrees (from -90 to 90)
    - Distances from front and side walls
  - Current speed of the car

- The neural network outputs:
  - An acceleration factor (from 0.0 to 1.0)
  - A steering force (from -1.0 to 1.0)



Carwin


Maximize on Play
Stats
Gizmos
STATS
Speed : 15
Avg.Speed : 10
Distance : 15.67779
Best fitness: 11405
in generation: 3

Facts

- Fitness value = distanceMade x avgSpeed

- Our neural network is composed by
  1. Input layer with 5 neurons (one for each input)
  2. 1 hidden layer with 24 neurons (heuristically chosen)
  3. Output layer with 2 neurons (one for each output)

- The genetic population is a set of 14 chromosomes
- Remember that each neuron receives the total of weights from the upper level plus one "fake" weight for the bias!

How many weights in each chromosome?

(5 x 24) + (24 x 2) + 24 + 2 = 194!

DEMO

Fork it on 

This project is free and open-source.
Feel free to test and hack it!

https://github.com/alessandrofrancesconi/carwin