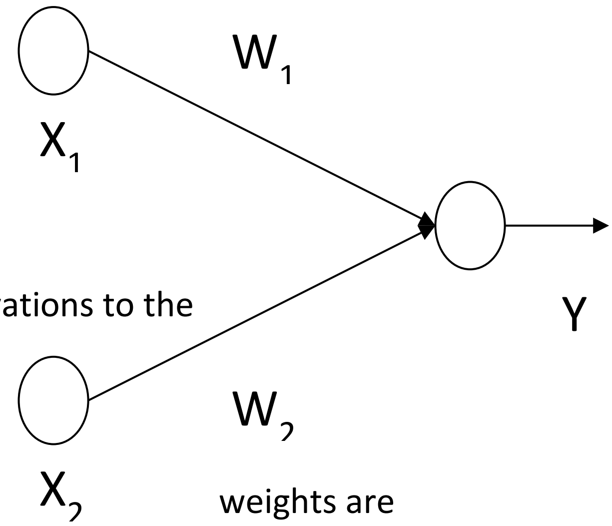
Genetic Algo:
Most often one is looking for the best solution in a specific subset of solutions. This subset is called the **search space** (or state space). Every point in the search space is a possible solution. Therefore every point has a **fitness** value, depending on the problem definition.
GAs are used to search the search space for the best solution, e.g. a minimum. Difficulties are the local minima and the starting point of the search.

Outline of the basic algorithm
0 START : Create random population of n chromosomes
1 FITNESS : Evaluate fitness f(x) of each chromosome in the population
2 NEW POPULATION
    1 REPRODUCTION/SELECTION : Based on f(x)
    2 CROSS OVER : Cross-over chromosomes
    3 MUTATION : Mutate chromosomes
3 REPLACE : Replace old with new population: the new generation
4 TEST : Test problem criterium
5 LOOP : Continue step 1 – 4 untill criterium is satisfied

Genetic Algorithm – Reproduction Cycle
1. Select parents for the mating pool
(size of mating pool = population size).
2. Shuffle the mating pool.
3. For each consecutive pair apply crossover.
4. For each offspring apply mutation (bit-flip independently for each bit).
5. Replace the whole population with the resulting offspring.

Crossover (Recombination):
Purpose: Crossover mixes the genes of two parent solutions to create new offspring.
Process: It takes parts from each parent and combines them to form one or more children.
Example: In a genetic algorithm for optimizing routes, it merges parts of two good routes to create a potentially better one.
Mutation:
Purpose: Mutation introduces random changes to solutions to keep things fresh.
Process: It tweaks a random element in a solution, adding a bit of randomness.
Example: In a genetic algorithm for scheduling, it might randomly swap two tasks to see if it leads to a better plan.

ARTIFICIAL NEURAL NET



- Information-processing system.
- Neurons process information.
- The signals are transmitted by means of connection links.
- The links possess an associated weight.
- The output signal is obtained by applying activations to the net input.
- The figure shows a simple artificial neural net with two input neurons (X_1, X_2) and one output neuron (Y). The interconnected weights are given by W_1 and W_2.

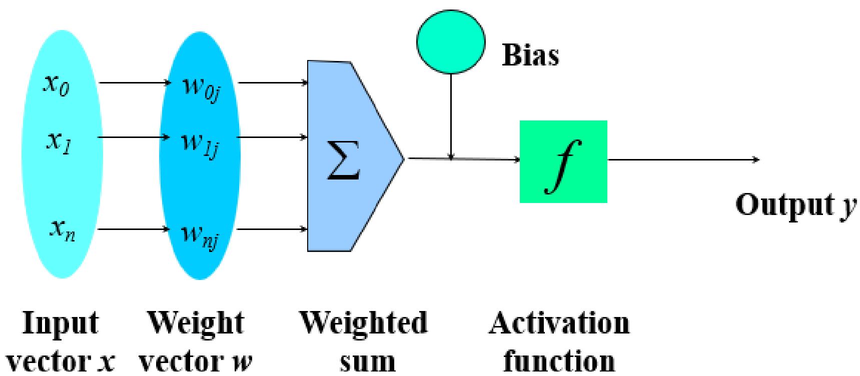
PROCESSING OF AN ARTIFICIAL NET

The neuron is the basic information processing unit of a NN. It consists of:

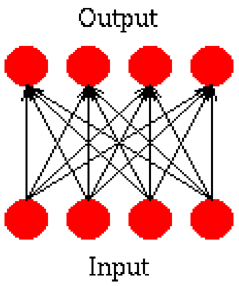
1. A set of links, describing the neuron inputs, with weights $W_1, W_2, \ldots, W_m$.
2. An adder function (linear combiner) for computing the weighted sum of the inputs (real numbers) $\mathbf{u} = \sum_{j=1}^{m} \mathbf{w}_j \mathbf{x}_j$
3. Activation function for limiting the amplitude of the neuron output.

$$y = \varphi(u + b)$$

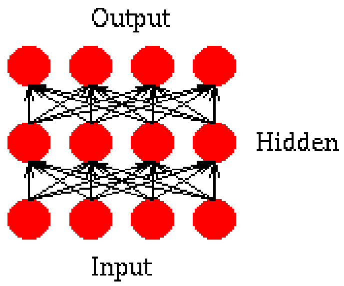
OPERATION OF A NEURAL NET



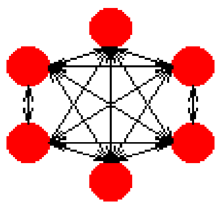
ARCITECTURE:



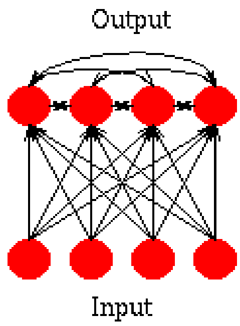
Single Layer Feedforward



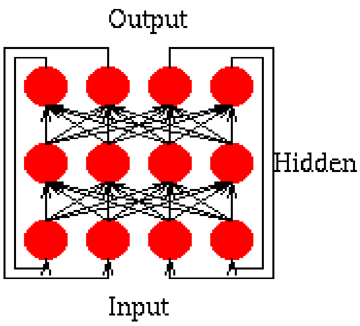
Multi Layer Feedforward



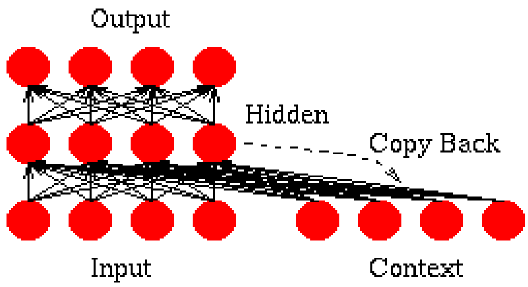
Fully Recurrent Network



Competitive Network

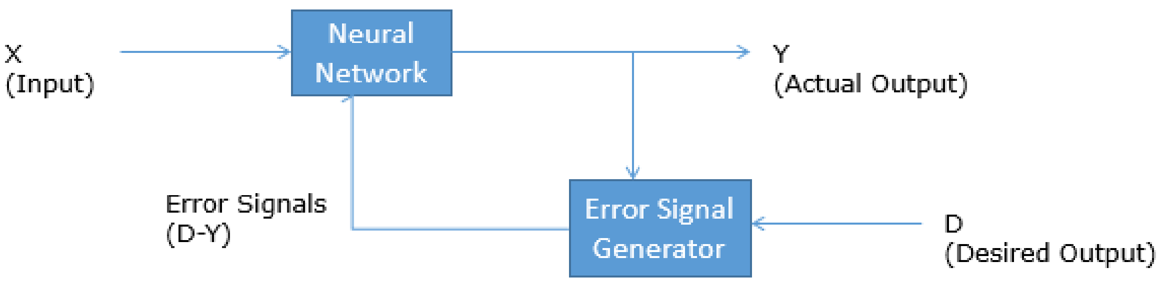


Jordan Network

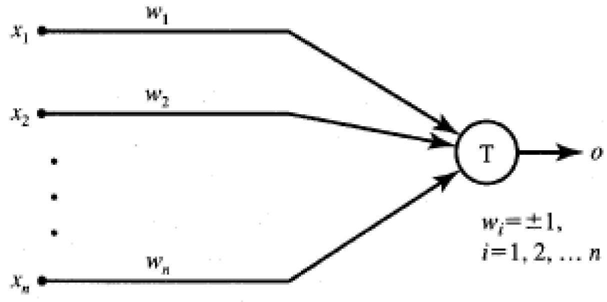


Simple Recurrent Network

Supervised Training - Providing the network with a series of sample inputs and comparing the output with the expected responses.



McCULLOCH–PITTS NEURON: First formal synthetic neuron model based on the highly simplified biological neuron



The inputs are 0 or 1
Outputs o is defined as

$$o^{k+1} = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_i x_i^k \geq T \\ 0 & \text{if } \sum_{i=1}^{n} w_i x_i^k < T \end{cases}$$

Though simplistic the model has sufficient computing potential
It can perform the basic logic operations NOT, OR, and AND, provided its weights and thresholds are appropriately selected

LEARNING AGENT:

Learning rule	Single weight adjustment Δw_{ij}	Initial weights	Learning	Neuron characteristics	Neuron / Layer
Hebbian	$co_i x_j$ $j = 1, 2, \ldots, n$	0	U	Any	Neuron
Perceptron	$c\left[d_i - \text{sgn}(\mathbf{w}_i^t \mathbf{x})\right] x_j$ $j = 1, 2, \ldots, n$	Any	S	Binary bipolar, or Binary unipolar*	Neuron
Delta	$c(d_i - o_i) f'(net_i) x_j$ $j = 1, 2, \ldots, n$	Any	S	Continuous	Neuron
Widrow-Hoff	$c(d_i - \mathbf{w}_i^t \mathbf{x}) x_j$ $j = 1, 2, \ldots, n$	Any	S	Any	Neuron
Correlation	$c d_i x_j$ $j = 1, 2, \ldots, n$	0	S	Any	Neuron
Winner-take-all	$\Delta w_{mj} = \alpha(x_j - w_{mj})$ m-winning neuron number $j = 1, 2, \ldots, n$	Random Normalized	U	Continuous	Layer of p neurons
Outstar	$\beta(d_i - w_{ij})$ $i = 1, 2, \ldots, p$	0	S	Continuous	Layer of p neurons

c, α, β are positive learning constants
S — supervised learning, U — unsupervised learning
* — Δw_{ij} not shown