A first proposed procedure to LEARN=TRAIN=CHANGE THE WEIGHTS (AND BIAS) TO IMPROVE PERFORMANCE, BY INTERACTING WITH DATA

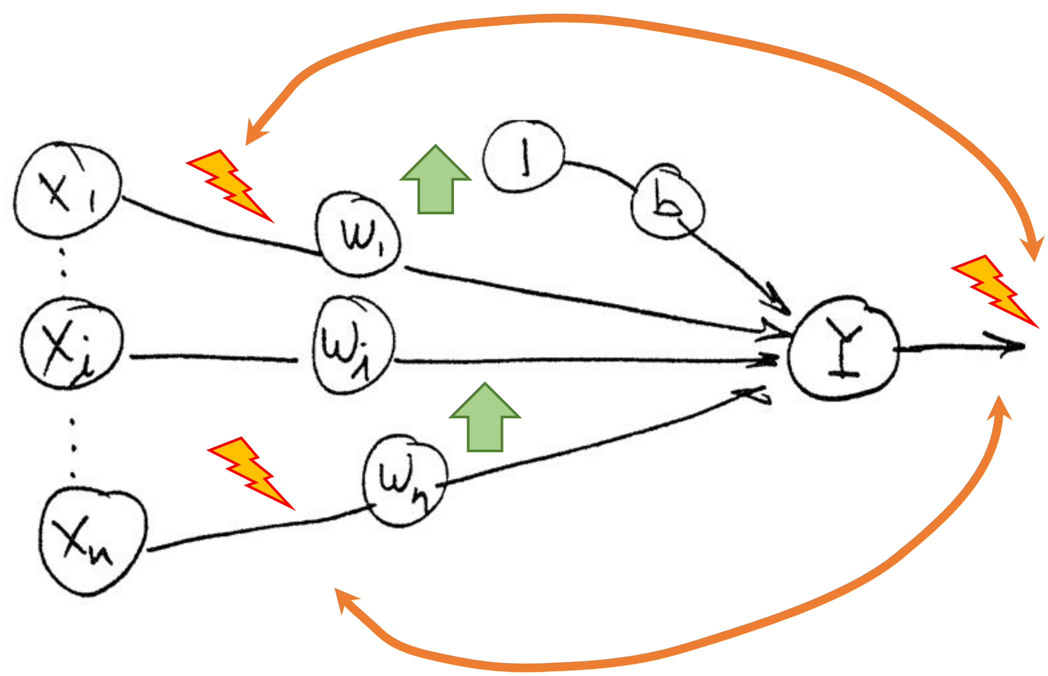
"Hebbian Learning"

Donald Hebb, a psychologist at McGill University, proposed in 1949 that learning takes place by modification of the synapse strengths (weights) in a manner such that ***if two interconnected neurons are both "on" at the same time, then the strength of the connection between those neurons (weight) should be increased.***

'Neurons that FIRE together, WIRE together'

IN ANN terms : w_i(new) = wi(old) + x_iy

The idea can be extended so that the connection is also enhanced if both neurons involved are "off" at the same time.



Hebbian Learning Algorithm:

0) Initialize weights e.g., $w_i = 0$

1) For each training vector-target pair, do 2-4:

   2) Set activation for input units: $x_i = s_i$ (training vector)
   3) Set activation for output unit: $y = t$ (target)
   4) Adjust the weights and bias:
      - w_i (new) = w_i(old) + $x_i\, y$
      - b(new) = b(old) + y

If the weight and the inputs (including the bias) are handled as vectors, the weight upgrade can be calculated through a vector (matrix) equation:

$$\overline{w}\,(new) = \overline{w}(old) + \Delta\,\overline{w}$$

where the weight change vector $\Delta\overline{w}$ is:

$$\Delta\overline{w} = \bar{x}\, y$$

This is:

$$\overline{w}(new) = \overline{w}(old) + \bar{x}\, y$$

Example:

AND function with bipolar inputs and targets

The "Truth Table" for the AND function, including a bias term will be

INPUT			TARGET	Sample #
x_1	x_2	"1"(for bias)		
1	1	1	1	First
1	-1	1	-1	Second
-1	1	1	-1	Third
-1	-1	1	-1	Forth

Application of the algorithm above yields the following results:

Hebb Learning (AND bipolar inputs and targets)

Inputs			Target	Weight Changes			Weights		
x_1	x_2	"1"		Δw_l	Δw_2	Δb	w_1	w_2	b
							0	0	0
1	1	1	1	1	1	1	1	1	1
1	-1	1	-1	-1	1	-1	0	2	0
-1	1	1	-1	1	-1	-1	1	1	-1
-1	-1	1	-1	1	1	-1	2	2	-2

Note that at each point the weights for inputs and bias determine a "decision boundary" and that it "moves" to effectively *separate one kind of inputs from the other.*

$$x_2 = -\frac{w_1}{w_2} x_1 - \frac{b}{w_2}$$

Hebbian learning of bipolar "AND" function

$$x_2 = -\left(\frac{w_1}{w_2}\right)x_1 - \left(\frac{b}{w_2}\right) \Big\} \text{ Decision Boundary}$$