



HELP SLIDES

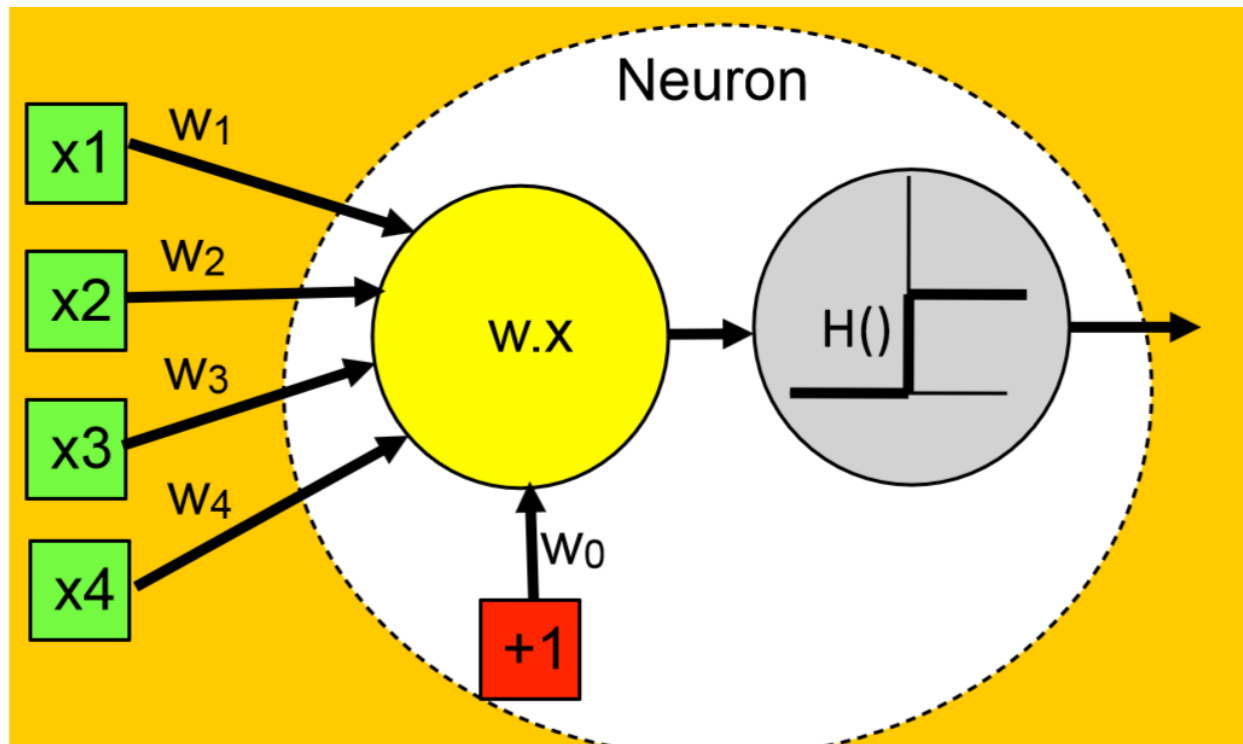


NEURAL NETWORK – SMALL TIP

- To simplify the topology in regards to the output layer consider making two perceptrons instead of one.
- E.g.
 - One to classify between Class 1 and Class 2
 - Another to classify between Class 2 and Class 3.
- If you do this, you will only need one output neurode (or node).
- Requires you to split data



PERCEPTRON/NEURAL NETWORK BASICS



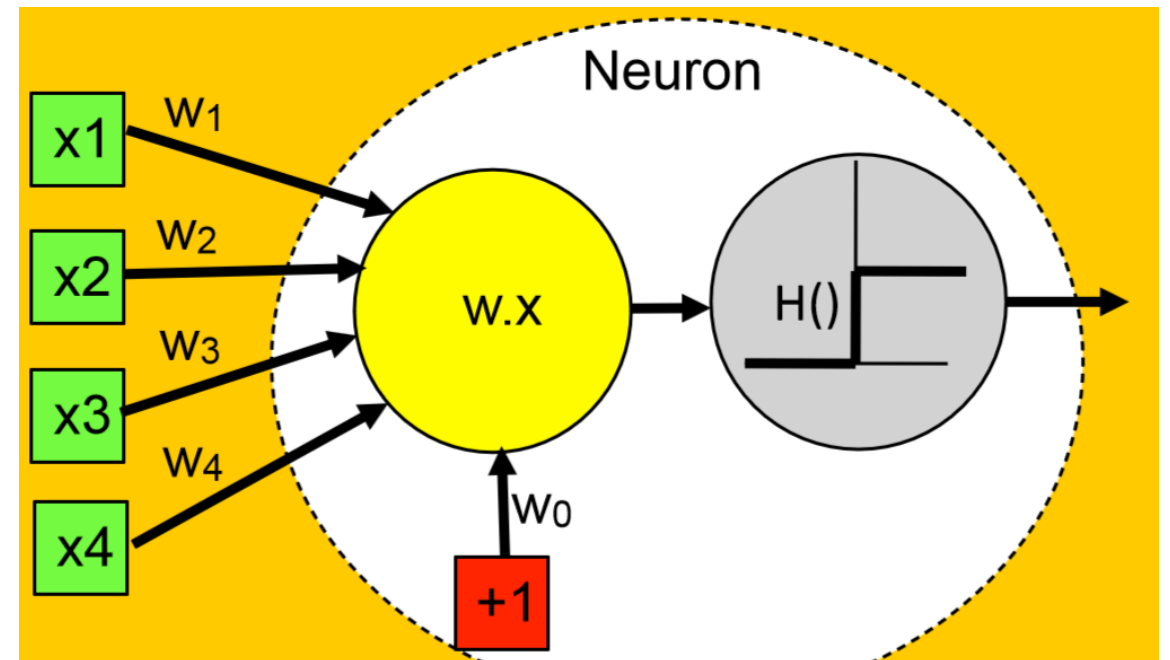
- Inputs (X)
- Connection Weights (W)
- Threshold/bias
- Weighted Sum
- Activation function
- Output

PERCEPTRON ALGORITHM

1. Initialize perceptron with random weights $[0 \dots 1]$ and bias value $[0 \dots 1]$, or with your own values (e.g. bias = 0)
2. For each set of inputs
 - Compute actual output, a^p , from perceptron using the activation function (more on next slide)
 - Update all weights with ΔW_j
3. If no changes to weights, then stop
4. Otherwise go back to 2

PERCEPTRON ALGORITHM – UPDATING WEIGHTS

- $\Delta W_j = \text{Learning rate} * X_j * (\text{desired output} - \text{actual output})$
- Example with W_1 , learning rate = 0.5, desired output = 0, actual output = 1:
- $\Delta W_1 = 0.5 * X_1 * (0 - 1)$



PERCEPTRON EXAMPLE

Threshold = Bias

Slide from a lecture given by Hector P. Martinez in the course *Modern AI for Games*, Fall 2012

Epoch	Inputs		Desired output d^p	Initial weights		Actual output a^p	Error E^p	Final weights	
	x_1	x_2		w_1	w_2			w_1	w_2
1	0	0	0	0.3	-0.1	0	0	0.3	-0.1
	0	1	0	0.3	-0.1	0	0	0.3	-0.1
	1	0	0	0.3	-0.1	1	-1	0.2	-0.1
	1	1	1	0.2	-0.1	0	1	0.3	0.0
2	0	0	0	0.3	0.0	0	0	0.3	0.0
	0	1	0	0.3	0.0	0	0	0.3	0.0
	1	0	0	0.3	0.0	1	-1	0.2	0.0
	1	1	1	0.2	0.0	1	0	0.2	0.0
3	0	0	0	0.2	0.0	0	0	0.2	0.0
	0	1	0	0.2	0.0	0	0	0.2	0.0
	1	0	0	0.2	0.0	1	-1	0.1	0.0
	1	1	1	0.1	0.0	0	1	0.2	0.1
4	0	0	0	0.2	0.1	0	0	0.2	0.1
	0	1	0	0.2	0.1	0	0	0.2	0.1
	1	0	0	0.2	0.1	1	-1	0.1	0.1
	1	1	1	0.1	0.1	1	0	0.1	0.1
5	0	0	0	0.1	0.1	0	0	0.1	0.1
	0	1	0	0.1	0.1	0	0	0.1	0.1
	1	0	0	0.1	0.1	0	0	0.1	0.1
	1	1	1	0.1	0.1	1	0	0.1	0.1

Threshold: $w_0 = -0.2$; learning rate $\eta = 0.1$