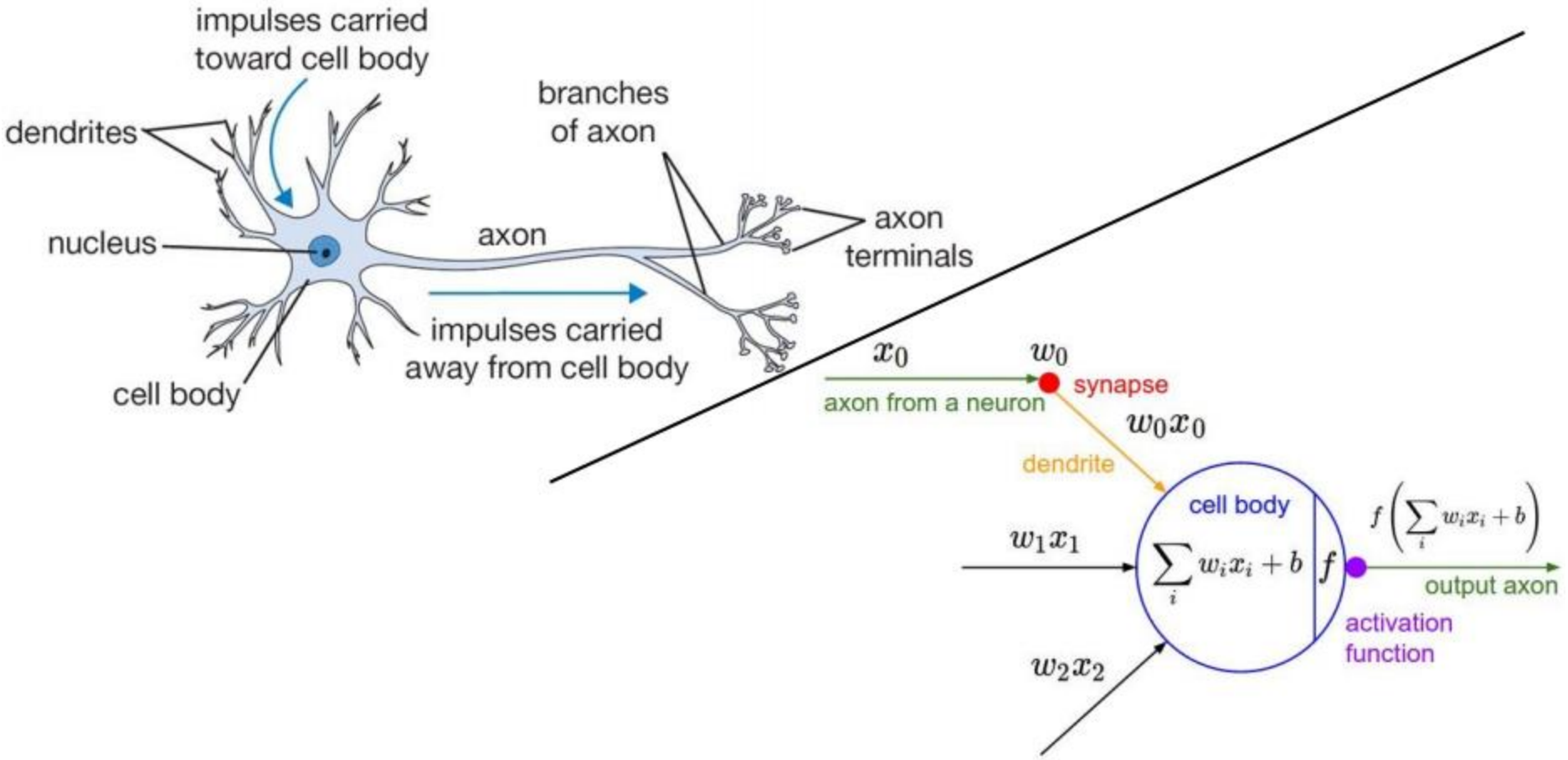
Perceptron, Neural Network

Eun-Sol Kim (김은솔)

Artificial Intelligence

Perceptron (1/3)

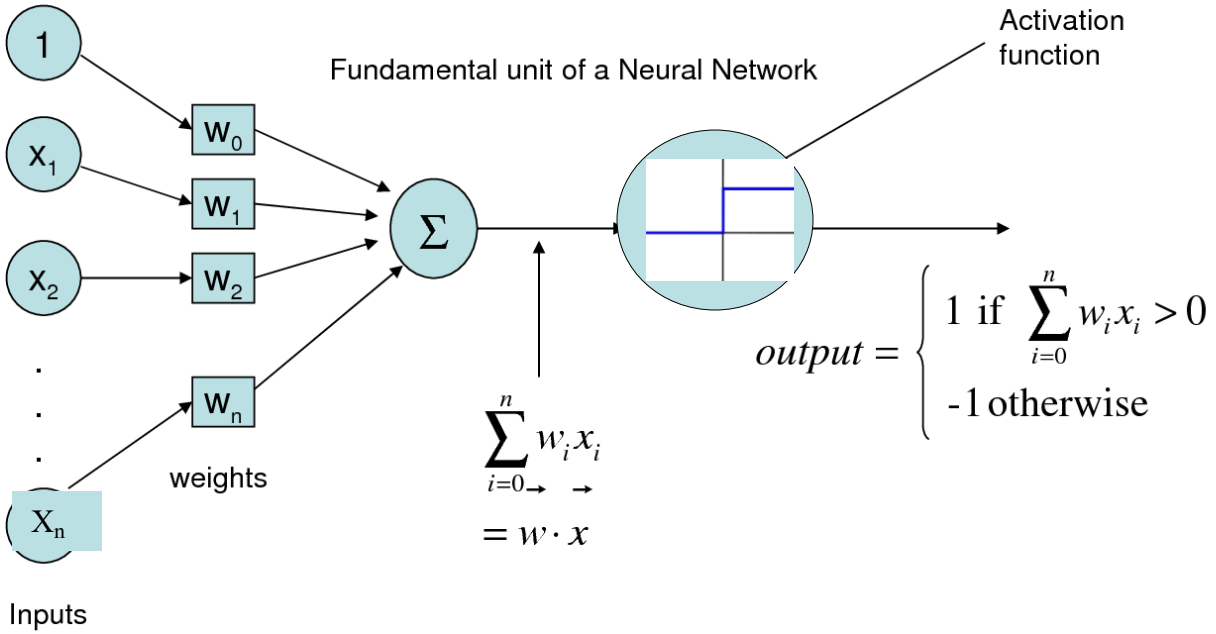


http://cs231n.stanford.edu/slides/winter1516_lecture4.pdf

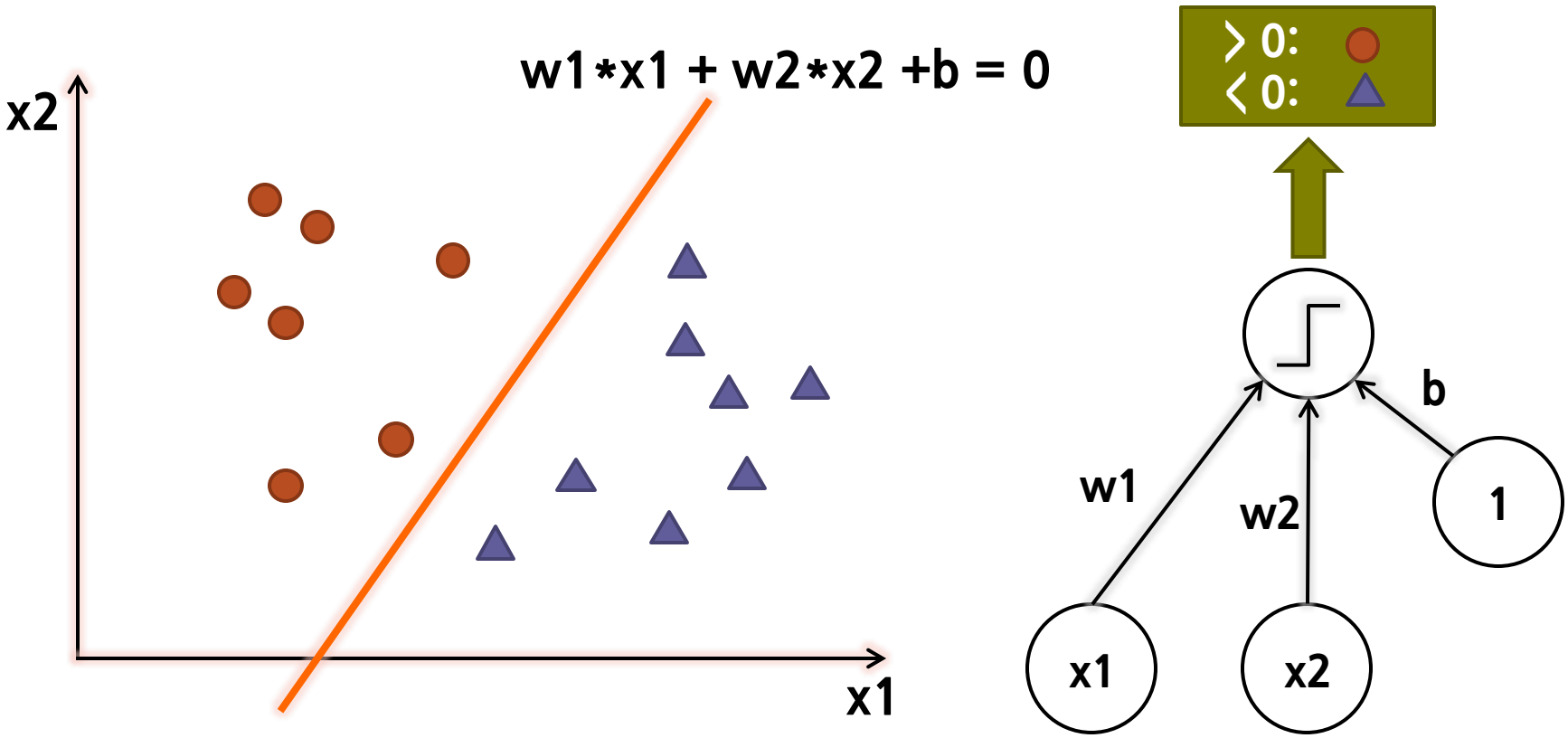
Perceptron (2/3)

Artificial Neural Networks

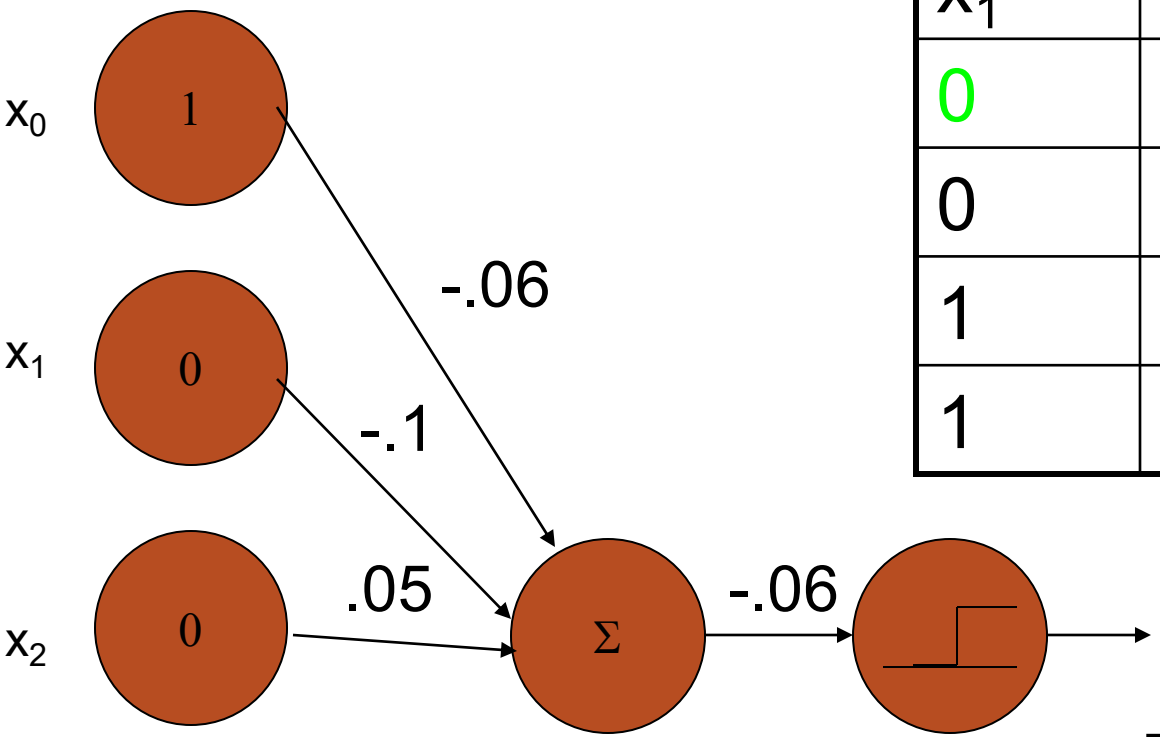
The Perceptron



Perceptron (3/3)

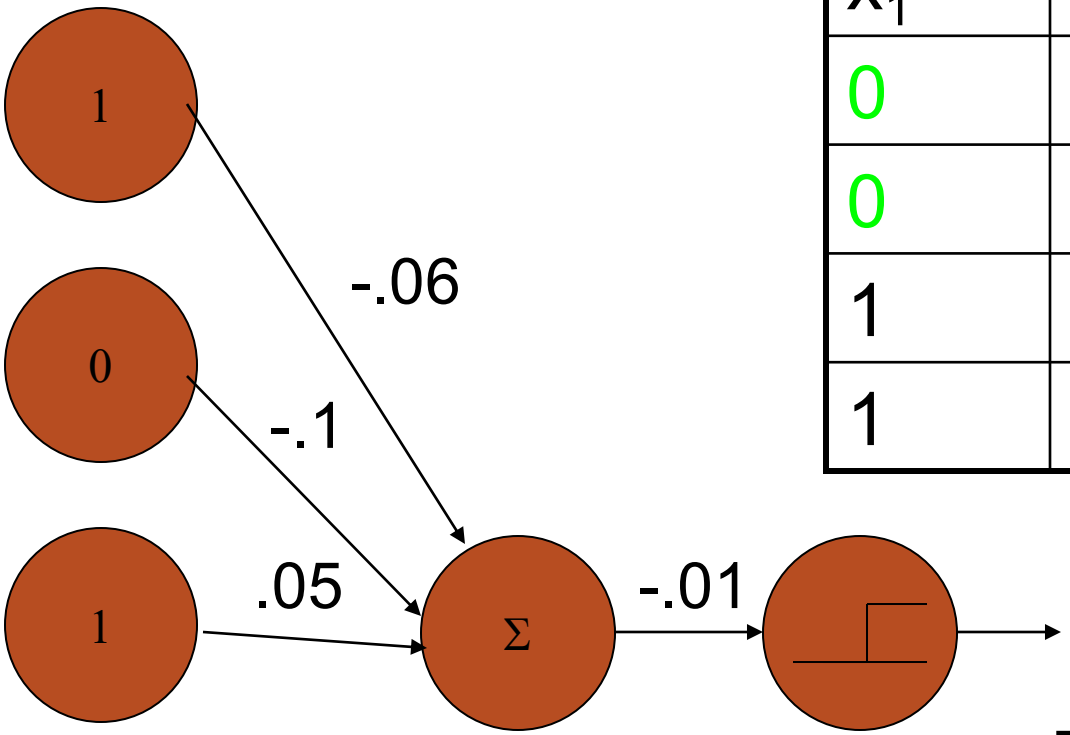


Perceptrons



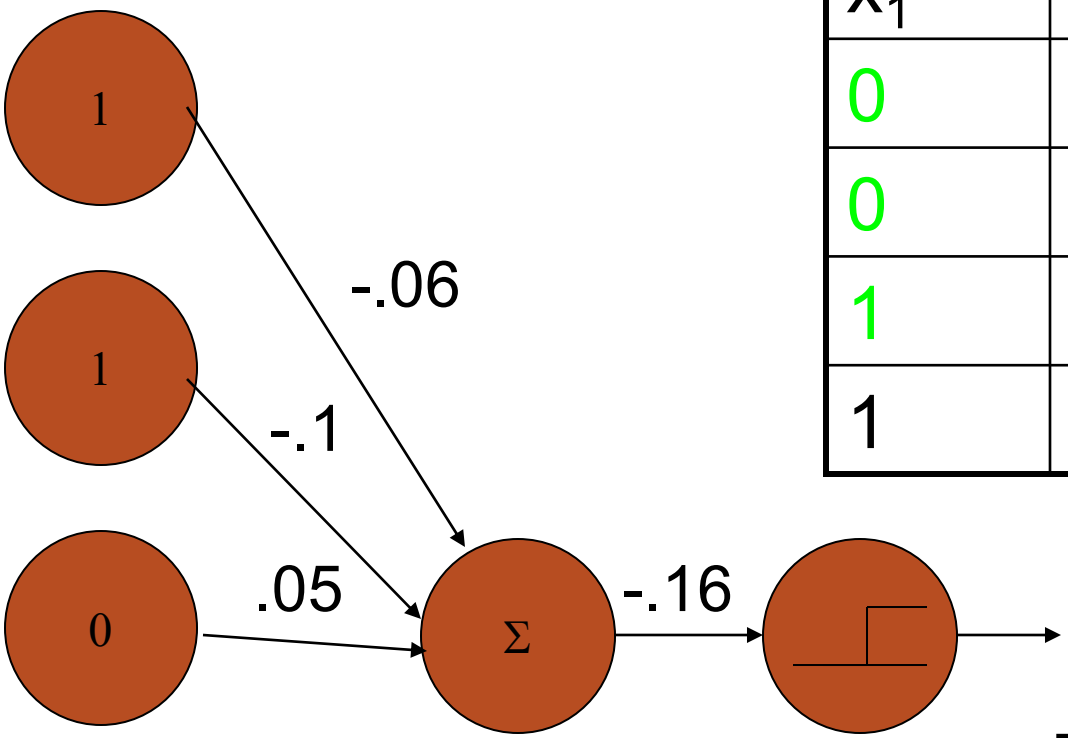
x_1	x_2	o
0	0	0
0	1	0
1	0	0
1	1	1

Perceptrons



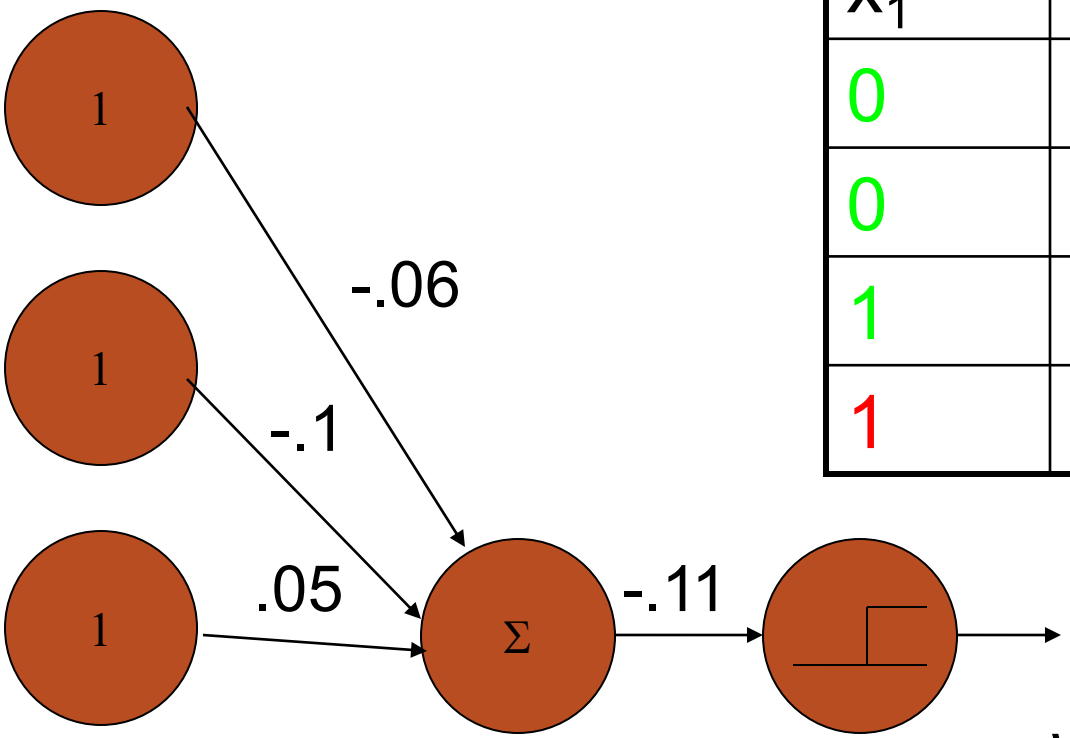
x_1	x_2	o
0	0	0
0	1	0
1	0	0
1	1	1

Perceptrons



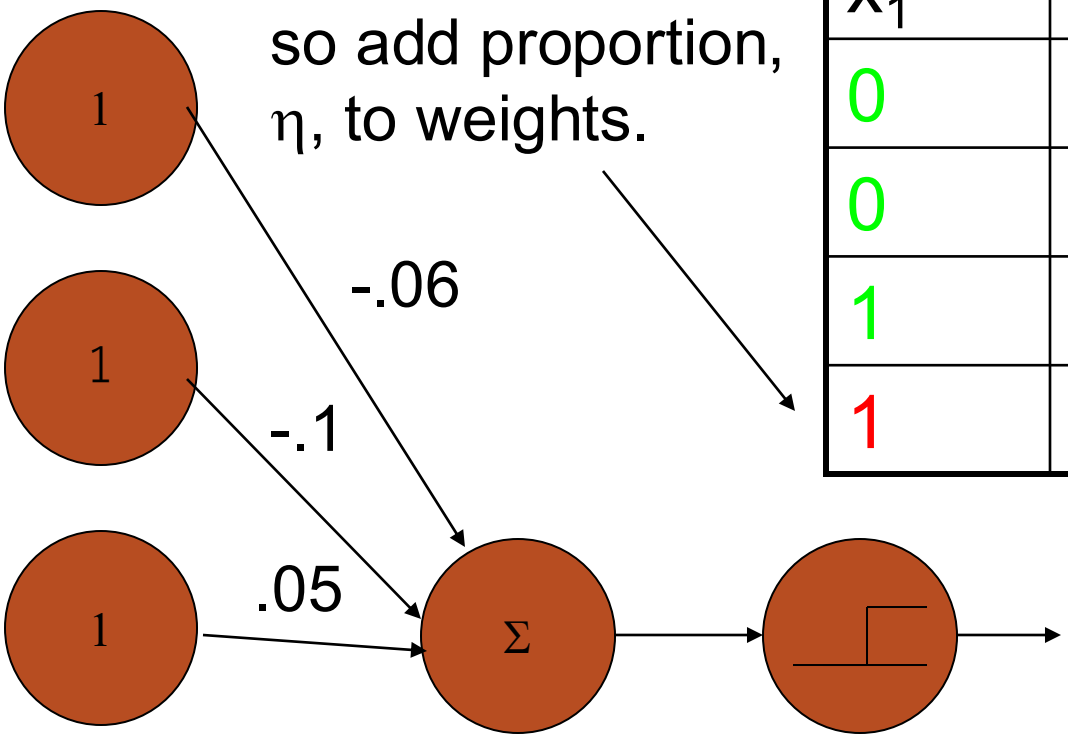
x_1	x_2	o
0	0	0
0	1	0
1	0	0
1	1	1

Perceptrons



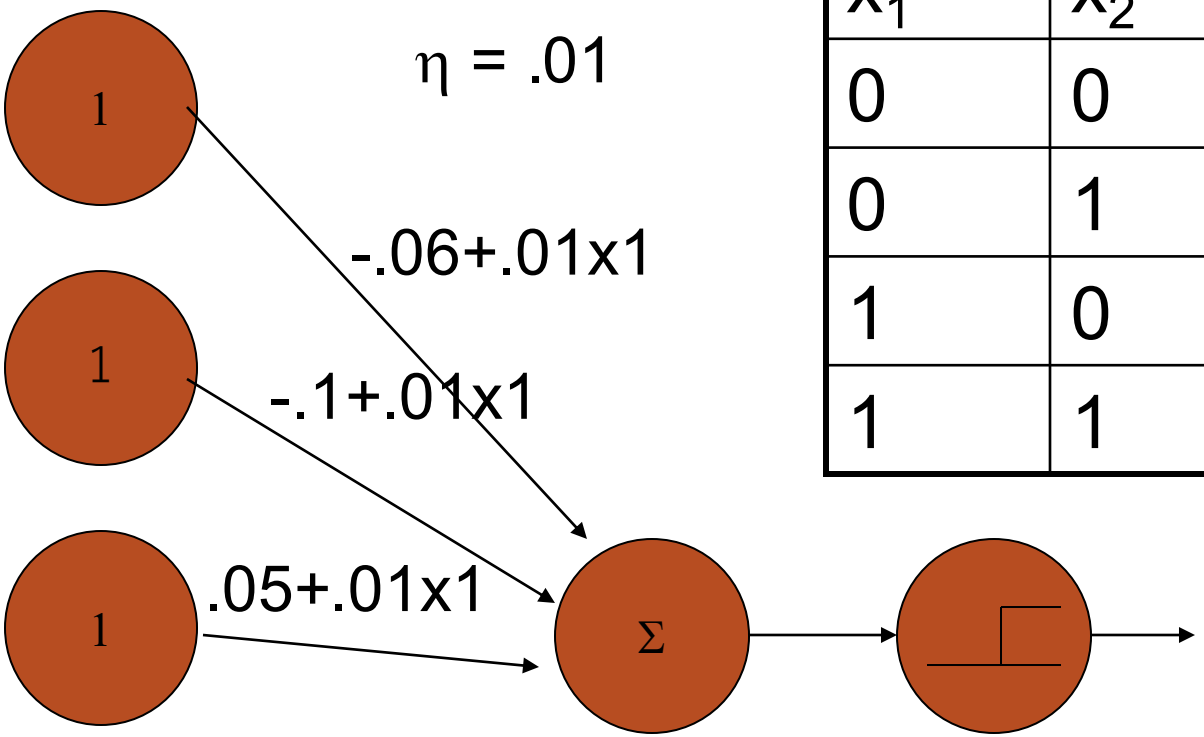
x_1	x_2	o
0	0	0
0	1	0
1	0	0
1	1	1

Perceptrons



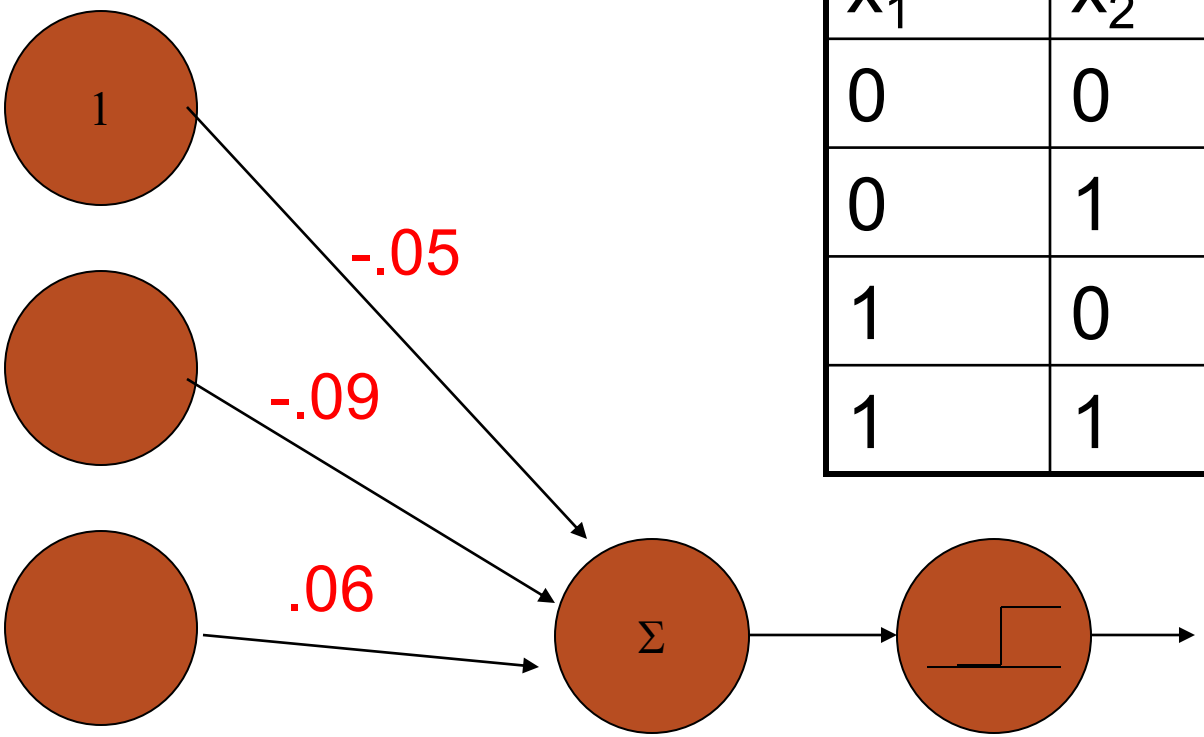
x_1	x_2	o
0	0	0
0	1	0
1	0	0
1	1	1

Perceptrons



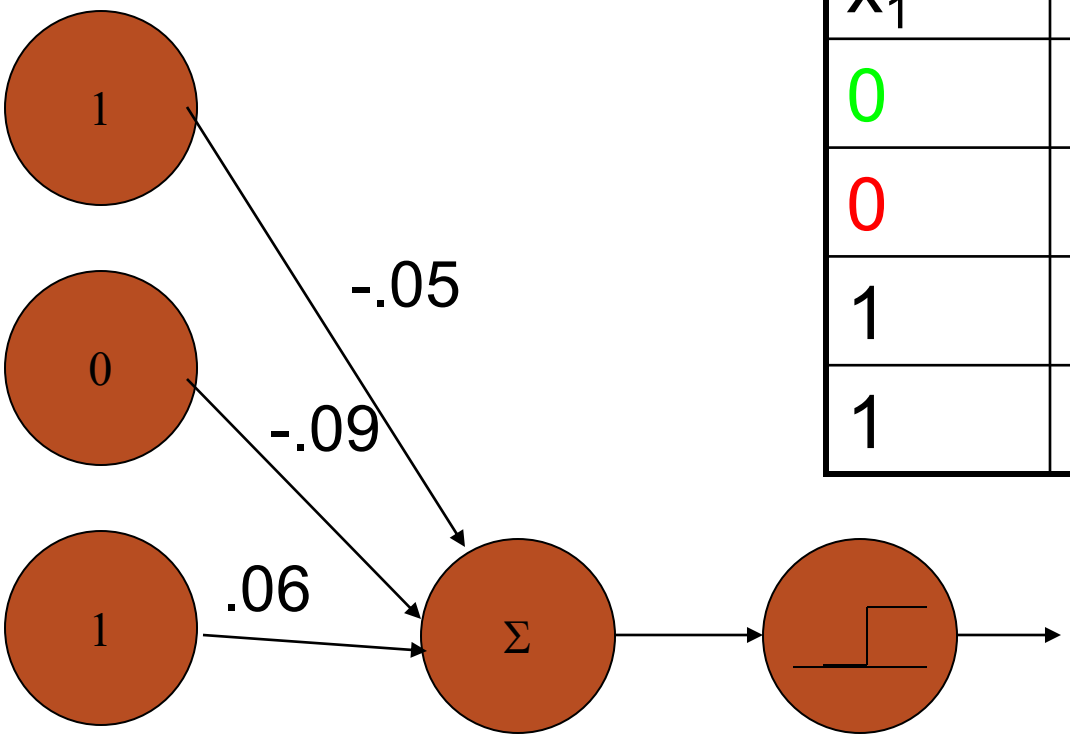
x_1	x_2	o
0	0	0
0	1	0
1	0	0
1	1	1

Perceptrons



x_1	x_2	o
0	0	0
0	1	0
1	0	0
1	1	1

Perceptrons

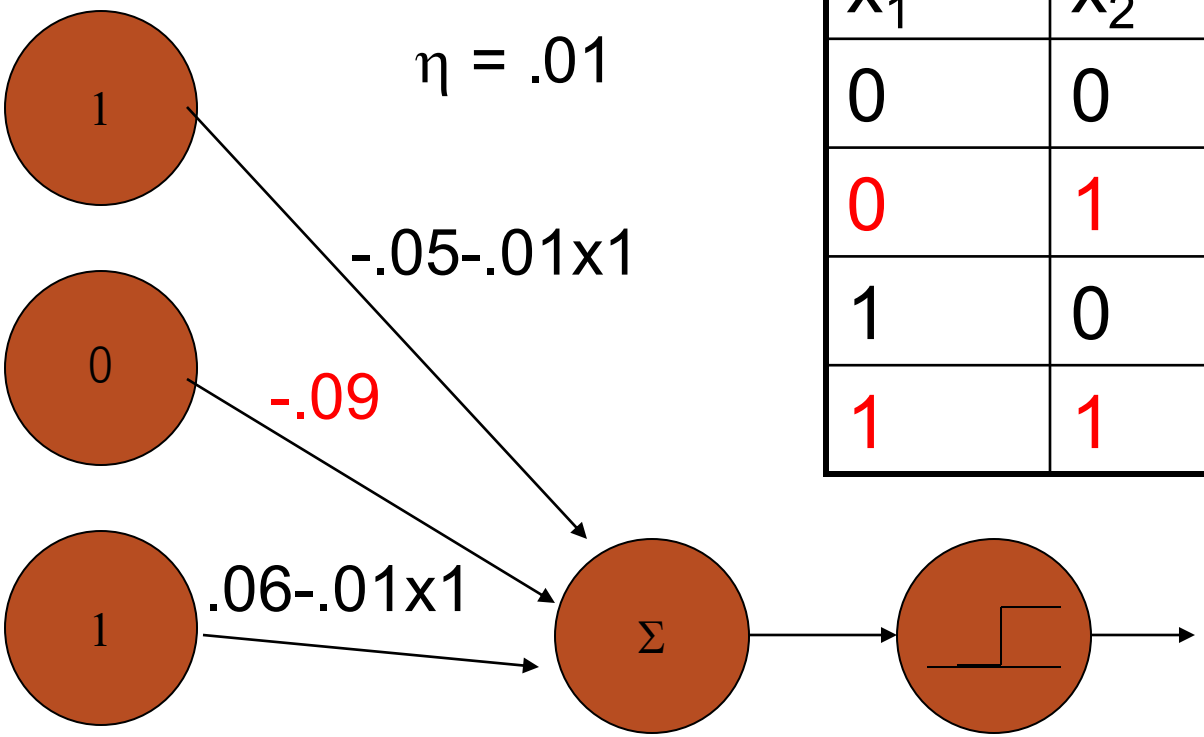


x_1	x_2	o
0	0	0
0	1	0
1	0	0
1	1	1

Decrease!

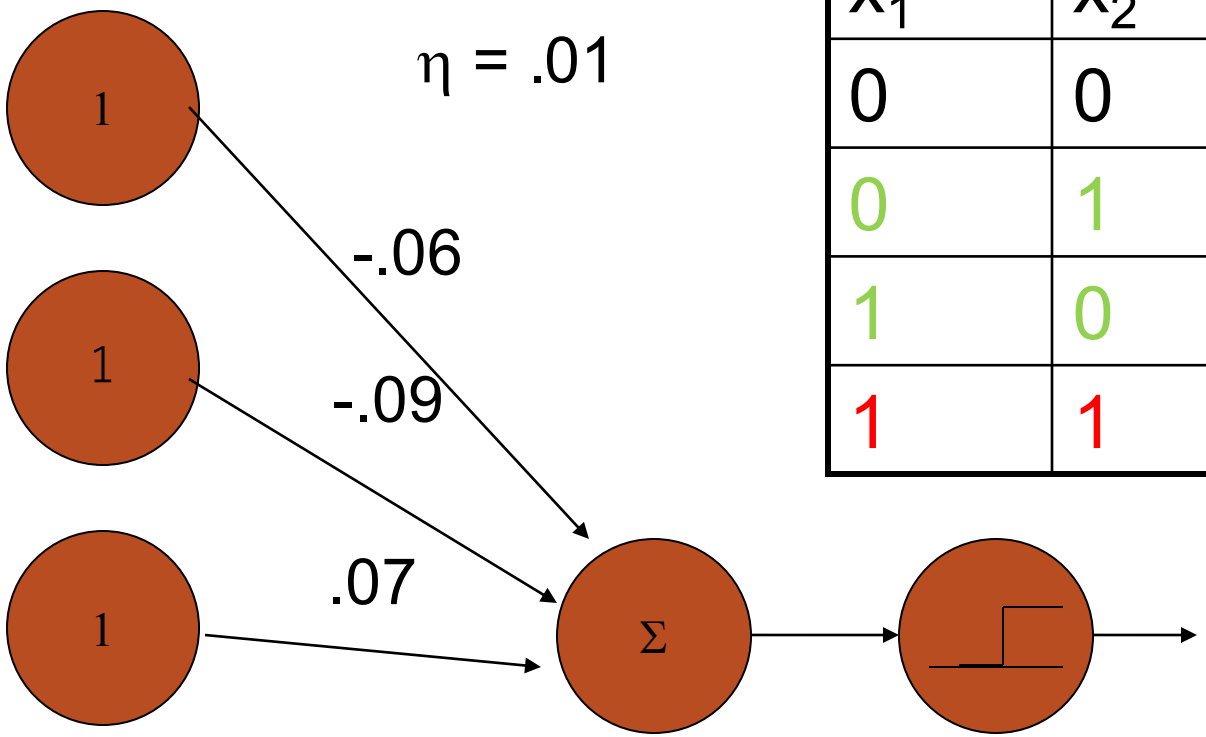
.01 1

Perceptrons



x_1	x_2	o
0	0	0
0	1	0
1	0	0
1	1	1

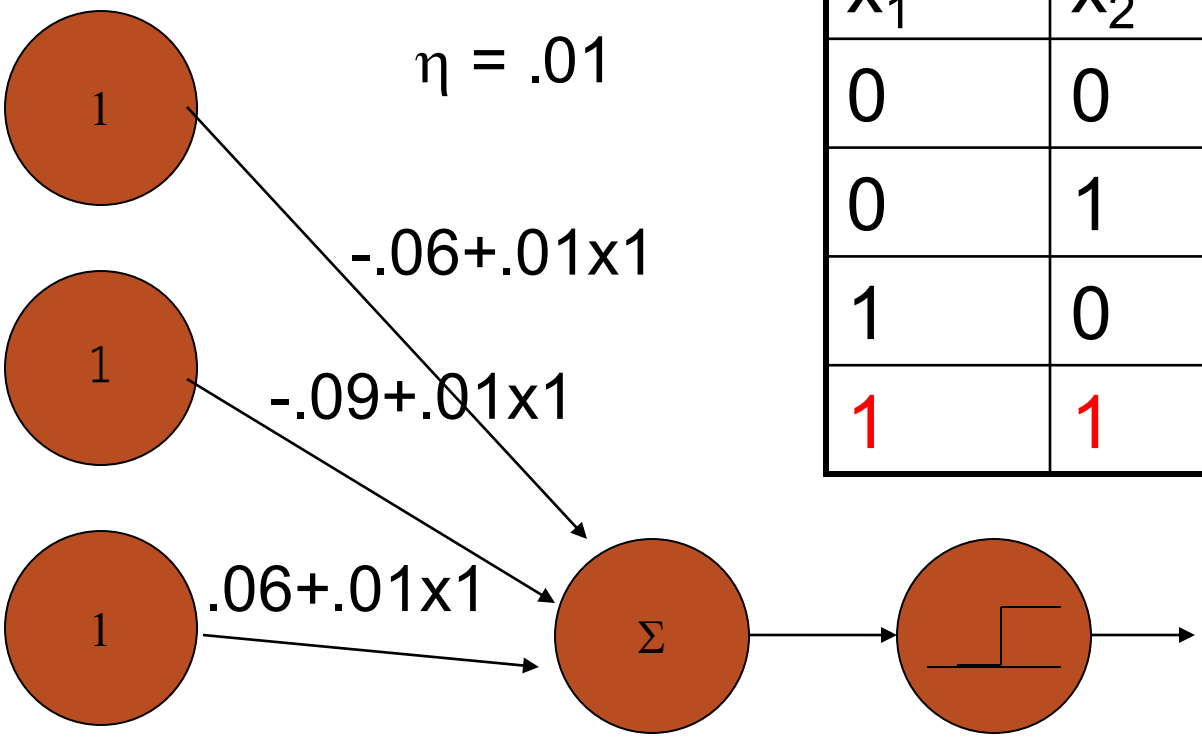
Perceptrons



x_1	x_2	o
0	0	0
0	1	0
1	0	0
1	1	1

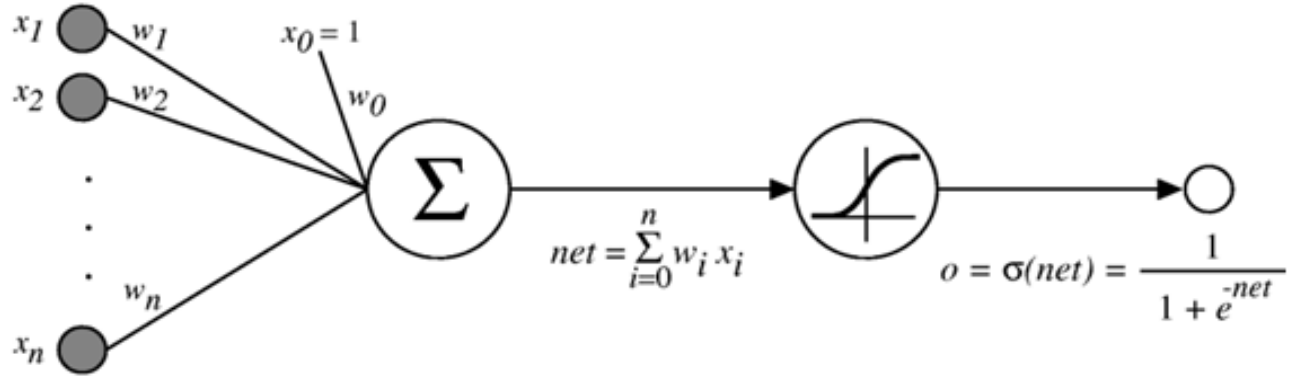
-.08

Perceptrons



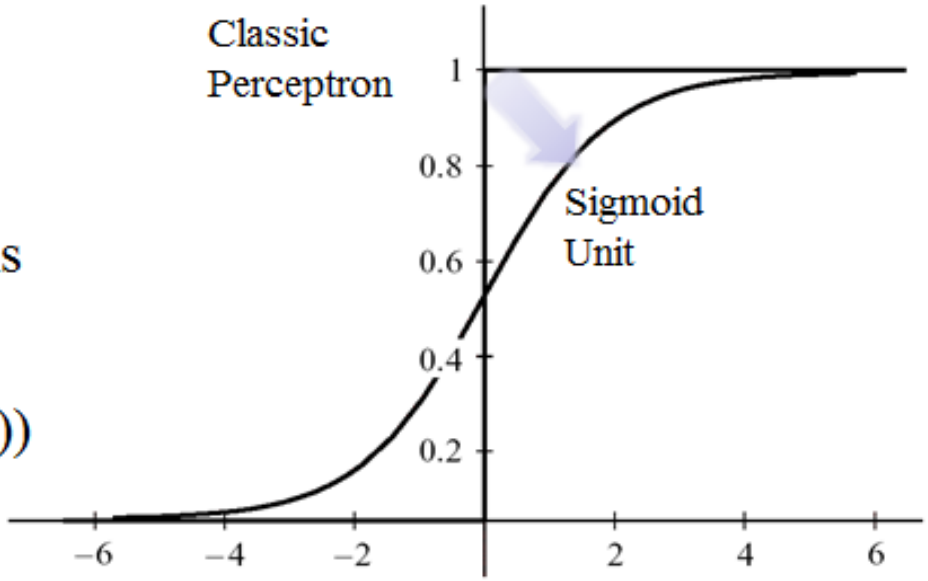
x_1	x_2	o
0	0	0
0	1	0
1	0	0
1	1	1

Sigmoid Unit



Sigmoid function is **Differentiable**

$$\frac{\partial \sigma(x)}{\partial x} = \sigma(x)(1 - \sigma(x))$$



Learning Algorithm of Sigmoid Unit

- Loss Function

Target (Label) — Model Output

$$\varepsilon = (d - f)^2$$

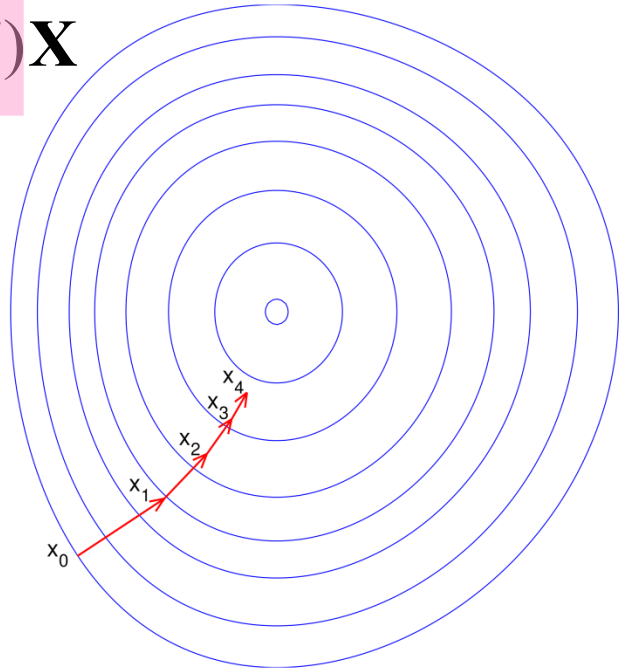
- Gradient Descent Update

$$\frac{\partial \varepsilon}{\partial \mathbf{W}} = -2(d - f)\frac{\partial f}{\partial s}\mathbf{X} = -2(d - f)f(1 - f)\mathbf{X}$$

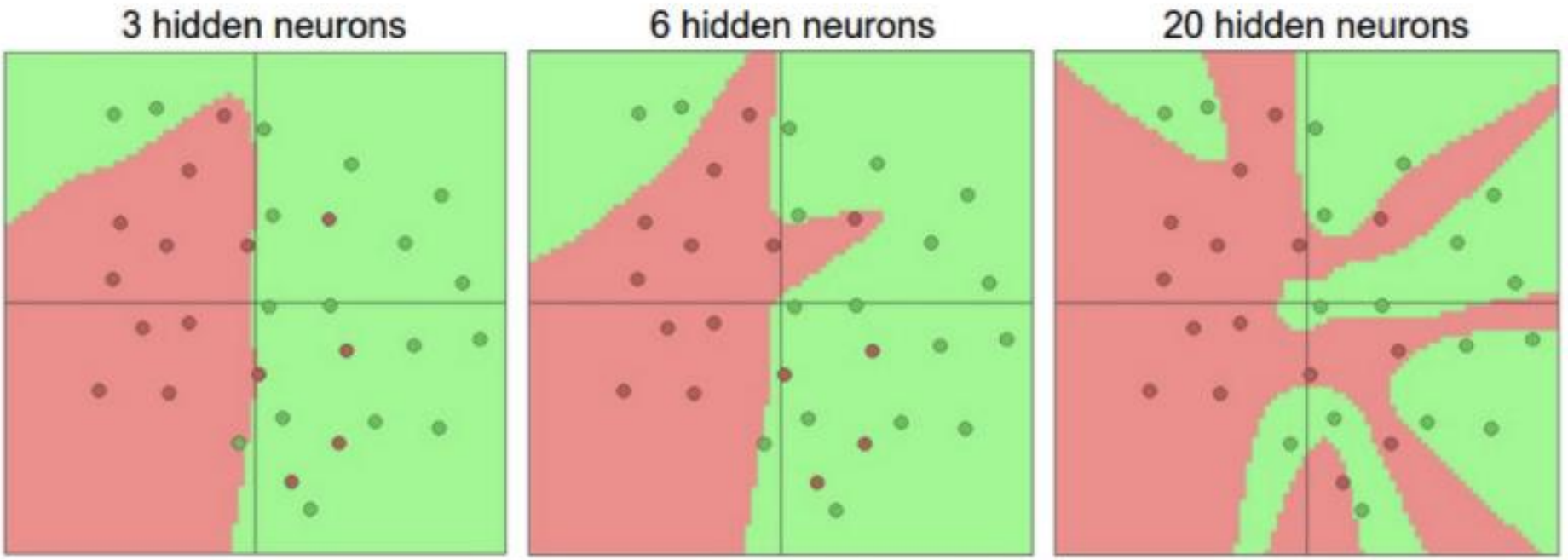
$$f(s) = 1/(1 + e^{-s})$$

$$f'(s) = f(s)(1 - f(s))$$

$$\mathbf{W} \leftarrow \mathbf{W} + c(d - f)f(1 - f)\mathbf{X}$$



Setting number of layers and their sizes



https://cs231n.github.io/neural-networks-1/

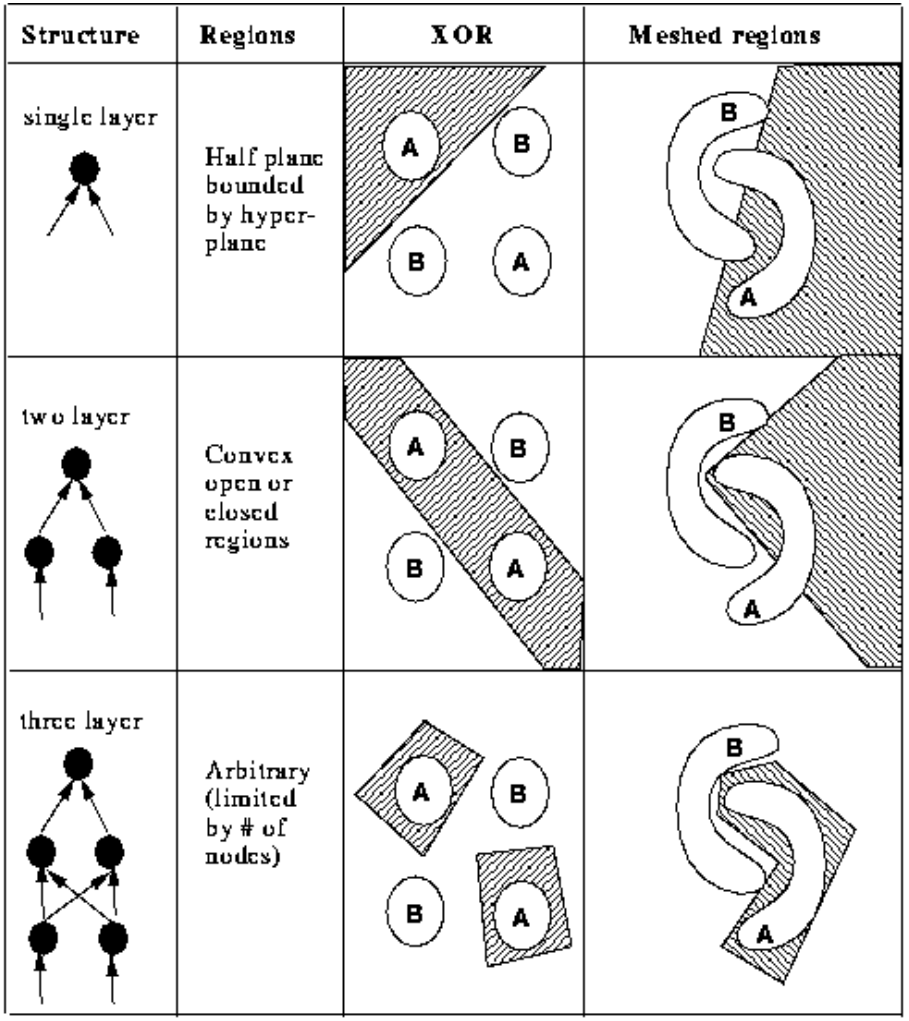
Need for Multiple Units and Multiple Layers

- Multiple boundaries are needed (e.g. XOR problem)

→ Multiple Units

- More complex regions are needed (e.g. Polygons)

→ Multiple Layers



http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html