Neural Network

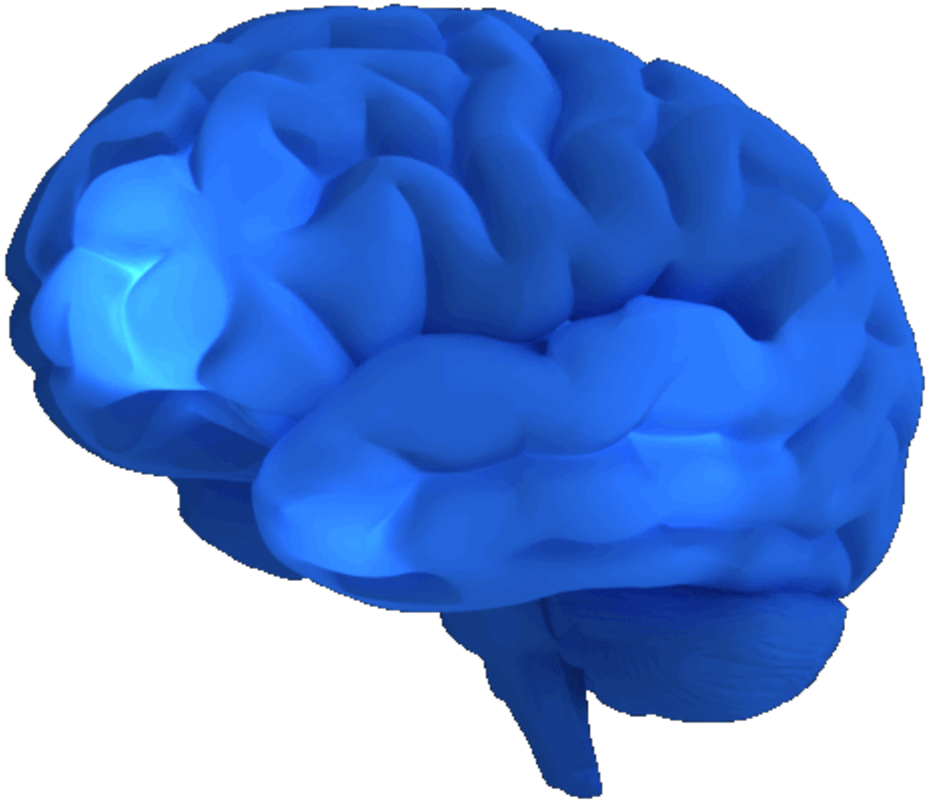
Anahita Zarei, Ph.D.

Overview

- Reading: Section 1.1.2 and e-chapter 7 from Learning from Data

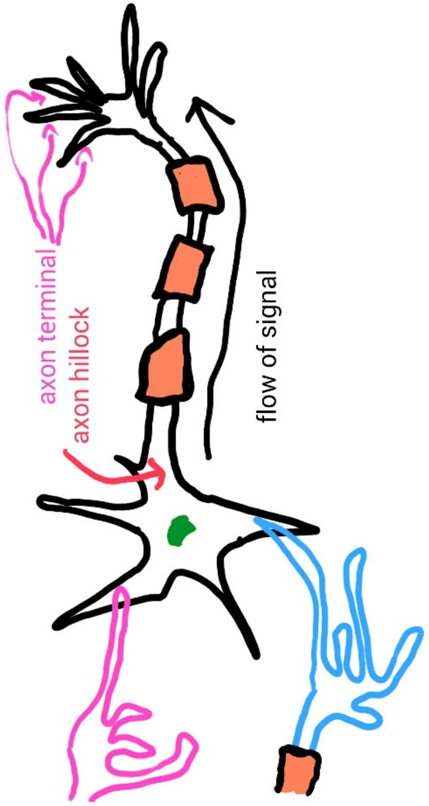
Why Neural Network?

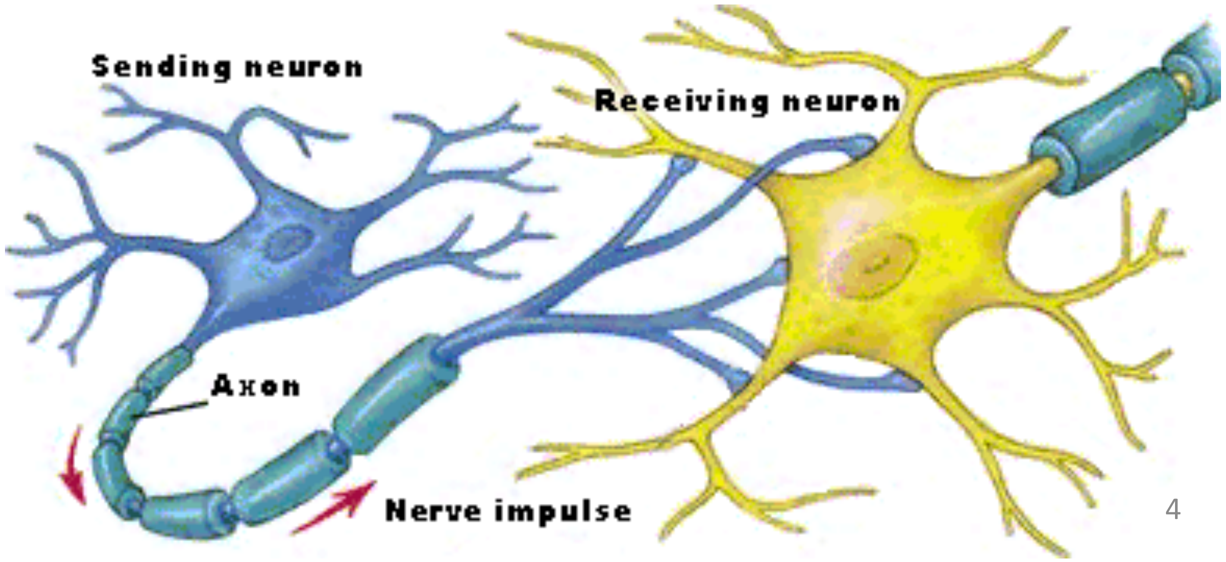
- It’s hip!
- It can learn complex functions from examples based on generalized training data in supervised learning.
- It can learn by extracting patterns and finding the underlying structure of the data in unsupervised learning.



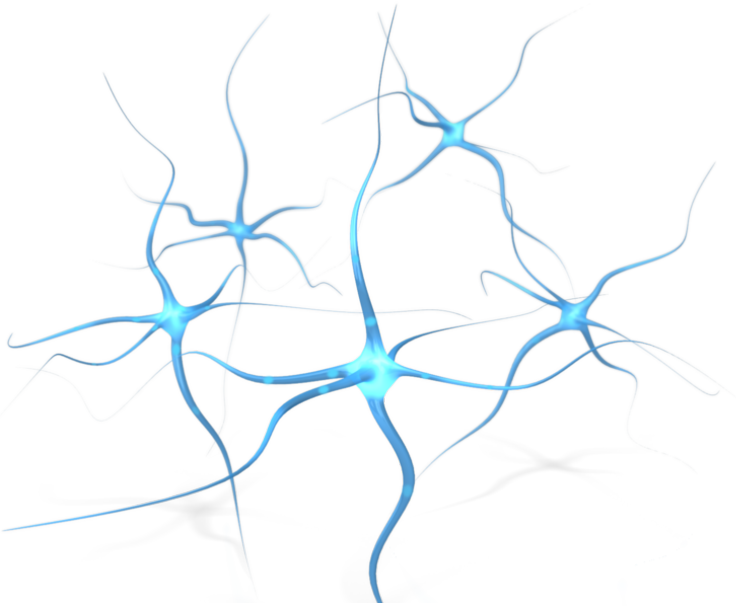
Human Neural Network

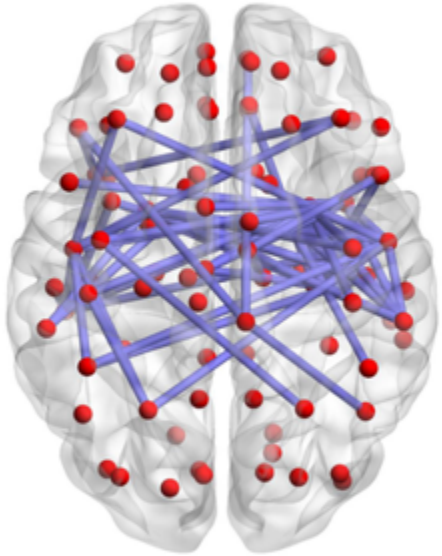
- Neuron is a fundamental cellular unit of the brain's nervous system.
- The biological neuron consists of four main parts:
  - Dendrites: resemble roots of a tree and act as input channels that receive impulses through synapses of other neurons.
  - Cell Body: processes (integrates) the signals received by the dendrites. If the combined input signal is strong enough the neuron "fires" .
  - Axon: resembles tree trunk. It conducts electrical impulses and transmit information to neighboring neurons.
  - Synapse: are gaps between neurons where neurons communicate with another. This junction is filled with neurotransmitter fluid.





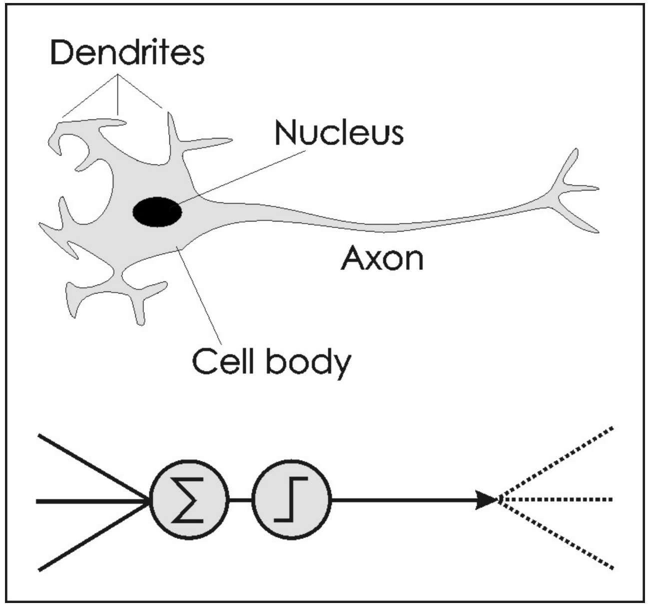
Human Neural Network

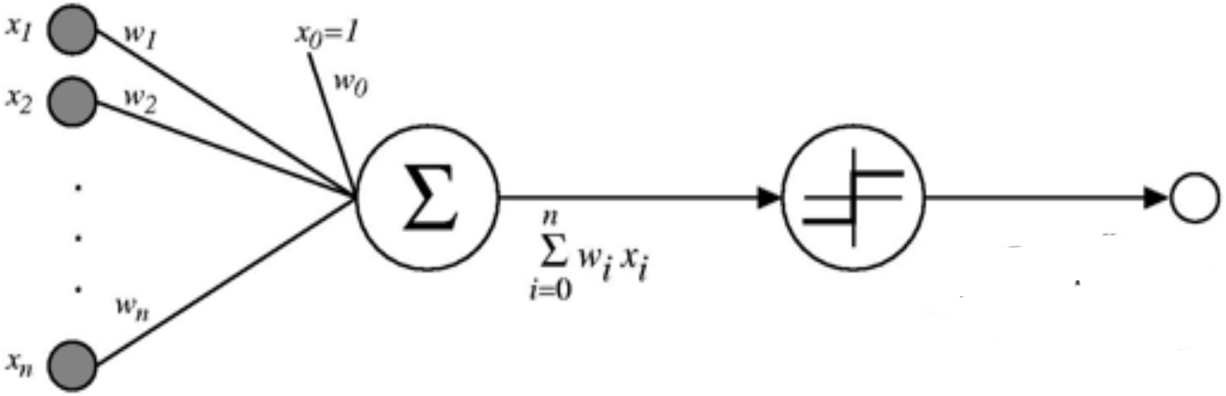




Artificial Neuron VS. Biological Neuron

Artificial Neuron	Biological Neuron
Lines that connect the input features to the summation processing element.	Dendrites
Processing element that has two parts: summation and the nonlinearity that decides if there’s an action potential or not.	Cell Body
The output of a neuron that is used by other neurons	Axon



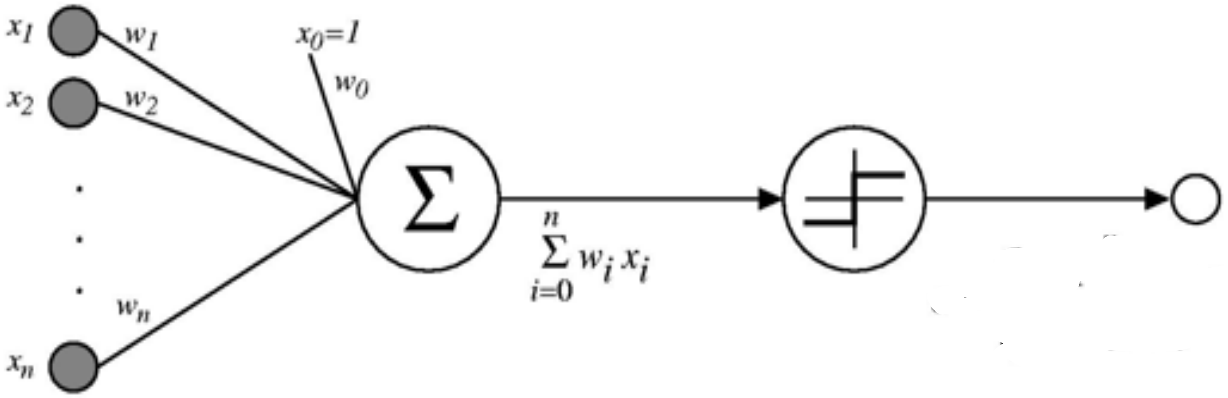


Single Neuron: Perceptron
Example: Approve or Deny Credit

[This guy is my hero!]

Single Neuron: Perceptron
Example: Approve or Deny Credit

For input $\mathbf{x} = (x_1, \cdots, x_d)$



Approve credit if $\sum_{i=1}^{d} w_i x_i > \text{threshold}$

Deny credit if $\sum_{i=1}^{d} w_i x_i < \text{threshold}$

$$h(\mathbf{x}) = \text{sign}\left(\left(\sum_{i=1}^{d} w_i x_i\right) - \text{threshold}\right)$$

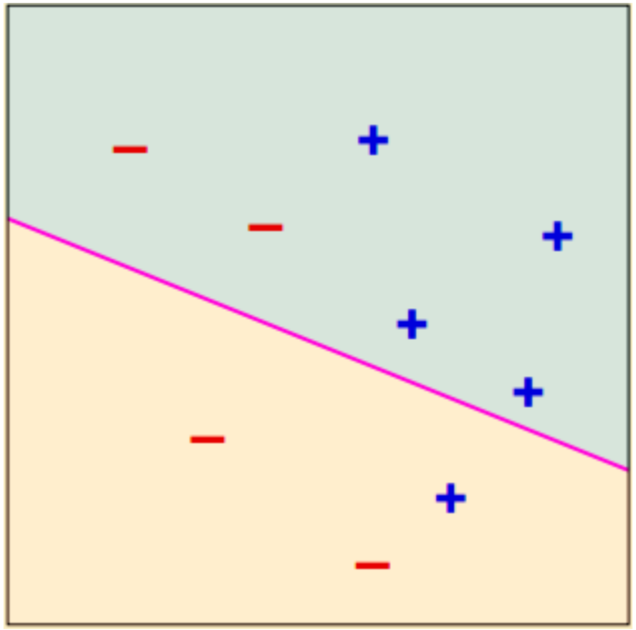
$$h(\mathbf{x}) = \text{sign}\left(\left(\sum_{i=1}^{d} w_i\, x_i\right) + w_0\right)$$

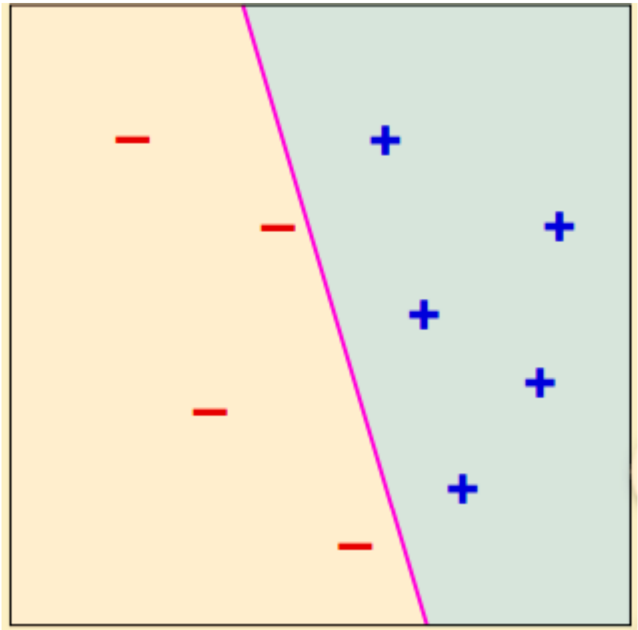
$$h(\mathbf{x}) = \text{sign}\left(\sum_{i=0}^{d} w_i\, x_i\right)$$

Perceptron Decision Boundary

If vector $\vec{X}$ is a row vector and vector $\vec{w}$ is a column vector then $h(\vec{X})$ can be expressed in terms of the dot products of these two vectors as follows:

$$h(\vec{X}) = sign(\vec{X}\vec{w})$$





Perceptron Learning Algorithm (PLA)

- PLA Pseudo Code:

a) Choose a misclassified point $(\overrightarrow{X_n}, y_n)$ from the following training set: $(\overrightarrow{X_1}, y_1) \dots (\overrightarrow{X_N}, y_N)$ where vector

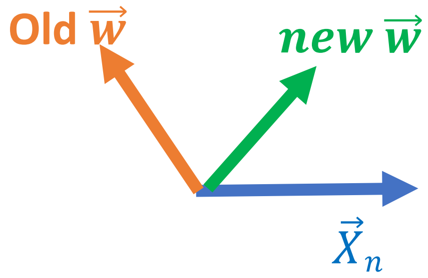
$$\vec{X} = (x_1, \dots, x_d)$$

b) Update the weight vector with the following rule

$$\boldsymbol{new\,\vec{w} = old\,\vec{w} + y_n\overrightarrow{X_n}}$$

c) Repeat the above process until all points are correctly classified.

Rational Behind PLA Algorithm

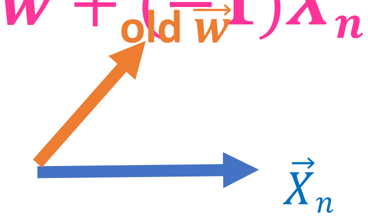


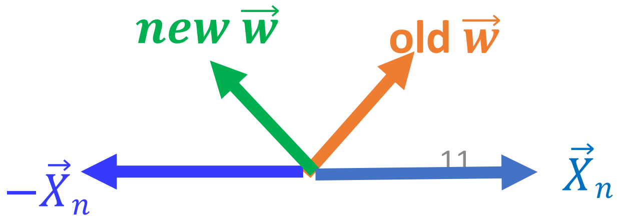
- Case 1: If the weight and X vector obtuse angle then the dot product will give you a negative value.
- $y_n = +1 \quad h(\vec{X}_n) = -1$

$$\boldsymbol{new\ \vec{w} = old\ \vec{w} + y_n \overrightarrow{X_n}}$$
$$\boldsymbol{new\ \vec{w} = old\ \vec{w} + (+1)\overrightarrow{X_n}}$$

- Case 2: If the weight and X vector acute angle then the dot product will give you a positive value.
- $y_n = -1 \quad h(\vec{X}_n) = +1$

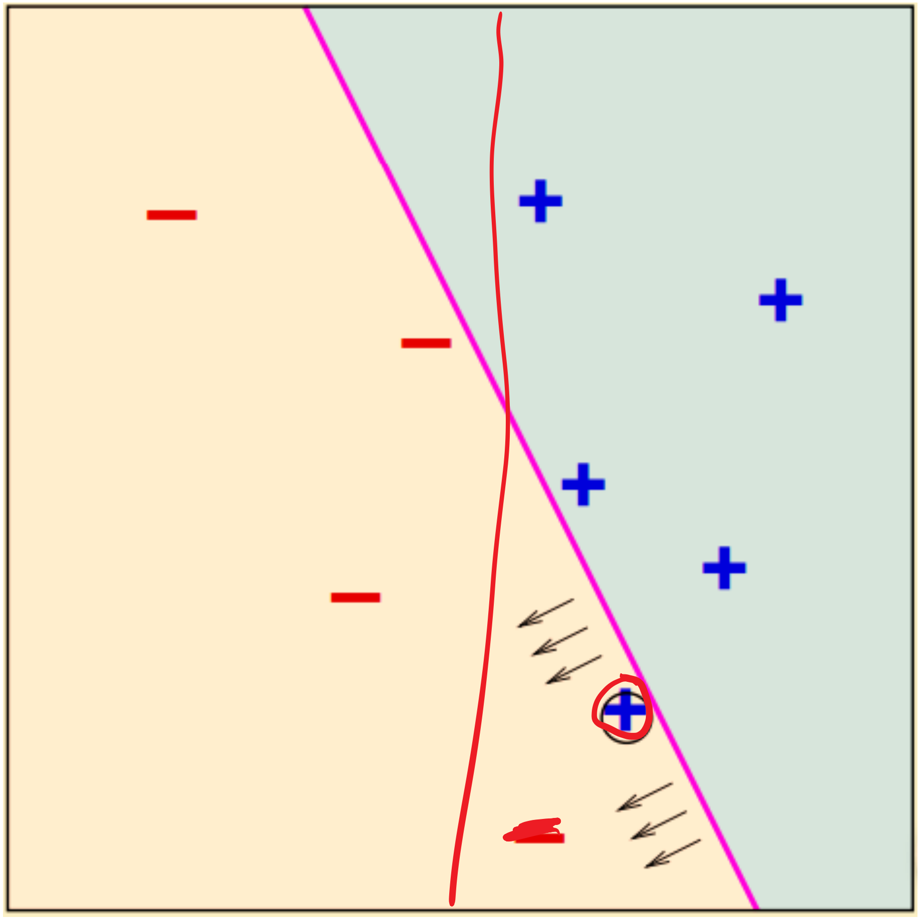
$$\boldsymbol{new\ \vec{w} = old\ \vec{w} + y_n \overrightarrow{X_n}}$$
$$\boldsymbol{new\ \vec{w} = old\ \vec{w} + (-1)\overrightarrow{X_n}}$$





PLA- cont.

- The update rule moves the boundary in the direction of classifying point X correctly, as showed in the figure.
- PLA considers only one training example at a time. In this process, it may misclassify some of the previously correctly classified points.
- However, it’s proved that there’s a guarantee that PLA converges to the correct boundary decision.
- Does this mean that this hypothesis will also be successful in classifying new data points that were not in the training set?

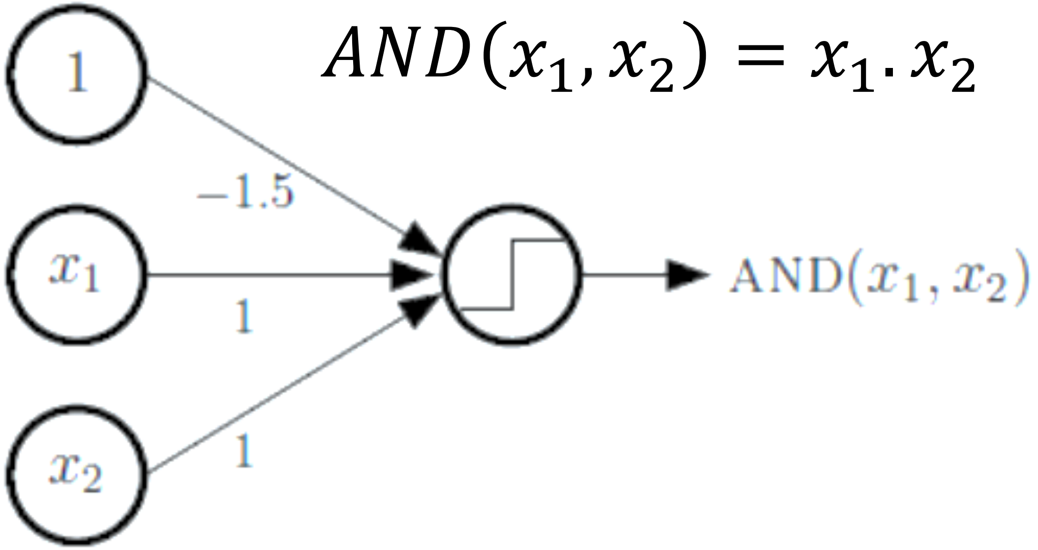


Example: Perceptron Implementation of an AND operator

- Recall the boundary equation:
- If $w_1x_1+w_2x_2$ > Threshold, System fires (y=1)
- If $w_1x_1+w_2x_2$ < Threshold, System doesn't fire (y=-1)
- If $w_1 = w_2 = 1$, what threshold value implements the AND operator?
  - -1-1 <Threshold => -2 < Threshold
  - 1+-1<Threshold => 0 < Threshold
  - 1+1 > Threshold => 2 >Threshold
- Any value between 0 and 2 would work for Threshold!
- For example if Threshold =1.5 then **w_0 = -1.5**
- $AND(x_1, x_2) = sign(x_1 + x_2 - 1.5)$.

x_1	x_2	$y = AND(x_1, x_2)=x_1 . x_2$
false (-1)	false (-1)	false (-1)
false (-1)	true (1)	false (-1)
true (1)	false (-1)	false (-1)
true (1)	true (1)	true (1)

$$AND(x_1, x_2) = x_1 . x_2$$

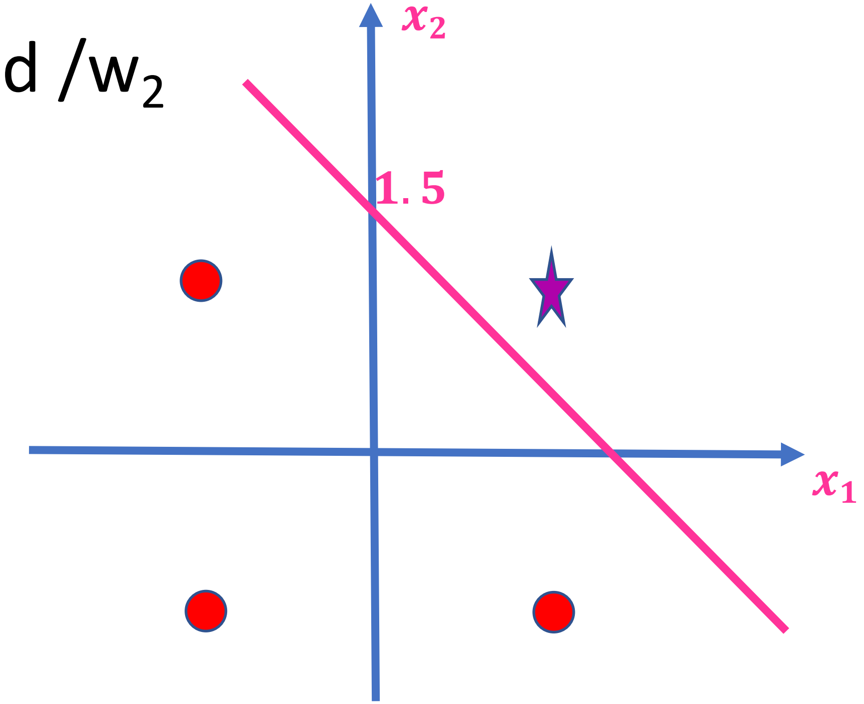


Example – cont.
Boundary Decision

x_1	x_2	$y = AND(x_1, x_2)$
false (-1)	false (-1)	false (-1)
false (-1)	true (1)	false (-1)
true (1)	false (-1)	false (-1)
true (1)	true (1)	true (1)

- $w_1 x_1 + w_2 x_2$ = Threshold, Dividing Line
- $x_2 = \frac{-w_1}{w_2} x_1 + \frac{Threshold}{w_2}$
- Slope = $-w_1/w_2$
- y-intercept = Threshold $/w_2$
- $x_2 = -x_1 + 1.5$

$$AND(x_1, x_2) = x_1 . x_2$$



Example: Perceptron Implementation of an OR operator

x_1	x_2	$y = OR(x_1, x_2) = x_1 + x_2$
false (-1)	false (-1)	false (-1)
false (-1)	true (1)	true (1)
true (1)	false (-1)	true (1)
true (1)	true (1)	true (1)

- Recall the boundary equation:
- $w_1x_1+w_2x_2$ > Threshold System fires (y=1)
- $w_1x_1+w_2x_2$ < Threshold System doesn't fire (y=-1)
- If $w_1 = w_2 = 1$, what threshold value implements the OR operator?
  - -1-1<Threshold => -2 < Threshold
  - 1-1 >Threshold => 0 > Threshold
  - 1+1 > Threshold => 2>Threshold
- Any value between 0 and -2 would work for Threshold!
- For example if Threshold=-1.5 then **w_0 = 1.5**
- $OR(x_1, x_2) = sign(x_1 + x_2 + 1.5)$.

$$OR(x_1, x_2) = x_1 + x_2$$