DECOMPOSING NEURAL NETWORKS

An applicant's guide to artificial learning
01.11.2022

→ JENNIFER MATTHIESEN & TINO PAULSEN | WINTERSEMESTER 2022

LEUPHANA
UNIVERSITÄT LÜNEBURG

RECAP
INTRODUCTION

- **MACHINE LEARNING** gives computers the ability to learn without being explicitly programmed
- ML requires **SAMPLES/DATA** and **FEATURES**
- **NEURAL NETWORKS** can find a non-linear decision boundary
- **SUPERVISED LEARNING**:
  - Class labels given
  - Like e.g. “The Teachable Machine”
- **UNSUPERVISED LEARNING**:
  - No class labels are given

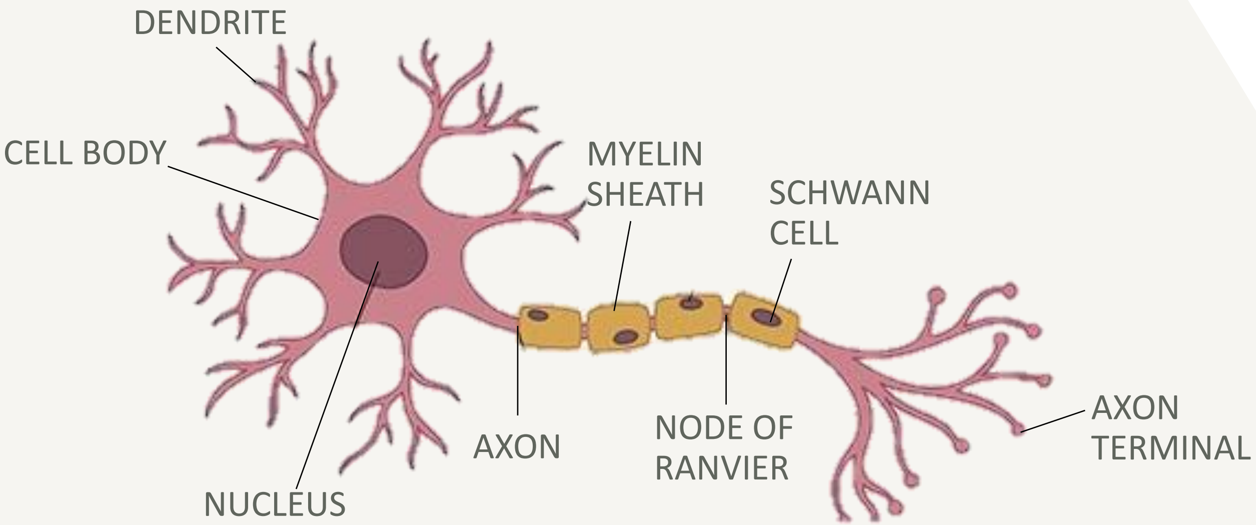
INTRO NN

- A biological neuron
- The artificial imitation: Perceptron
- Non-linear- decisions
- Cognitive psychology:
  - Geons
  - Bigram detectors
- Feature detection in images

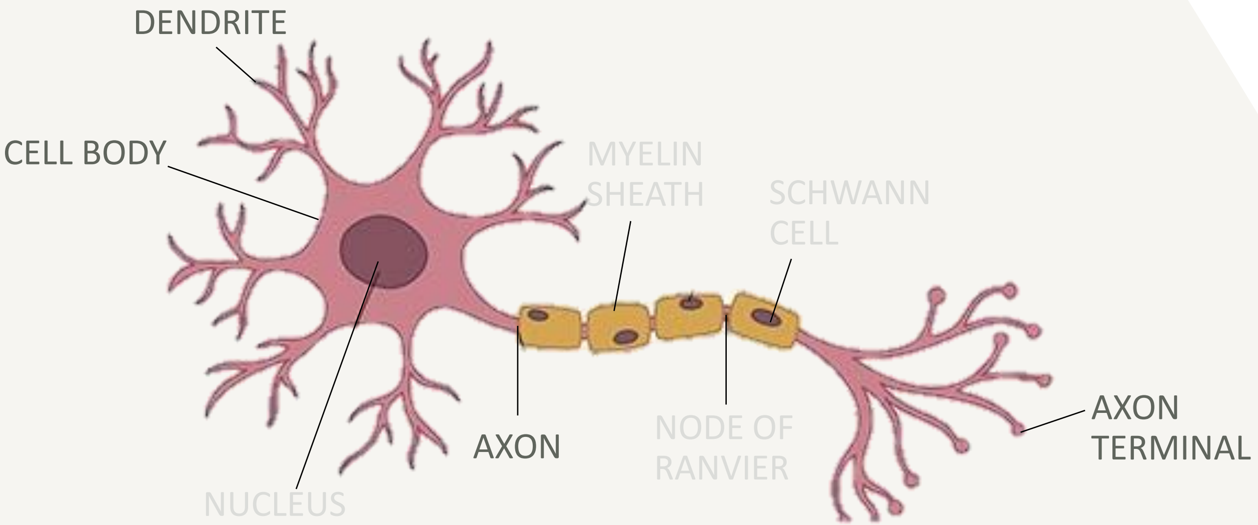
NEURAL NETWORK
WHY NEURAL COMPUTING?

- To understand how the brain actually works
  - The brain is big and complicated, and it gets damaged when you poke it.
- To solve practical problems by using novel learning algorithms inspired by the brain
  - Learning algorithms can be very useful even if they are not how the brain actually works.
- To investigate, how the imitation is different from the biological brain.
  - It is similar, better or worse?

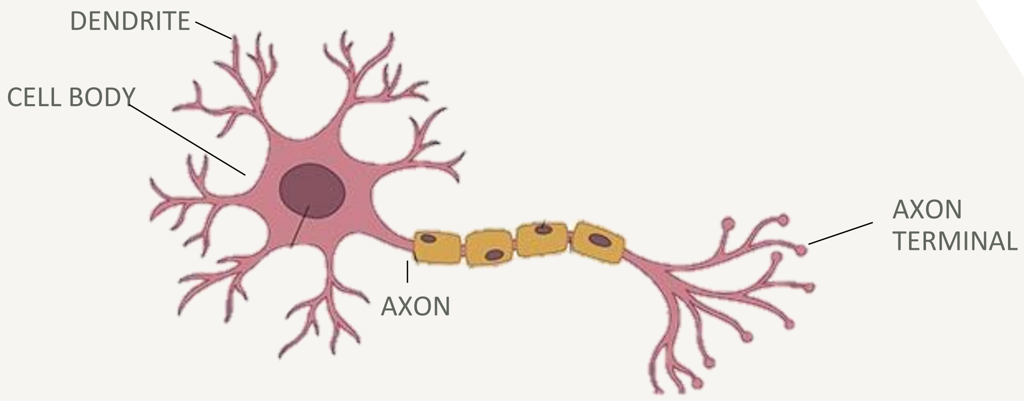
NEURAL NETWORK
A NEURON



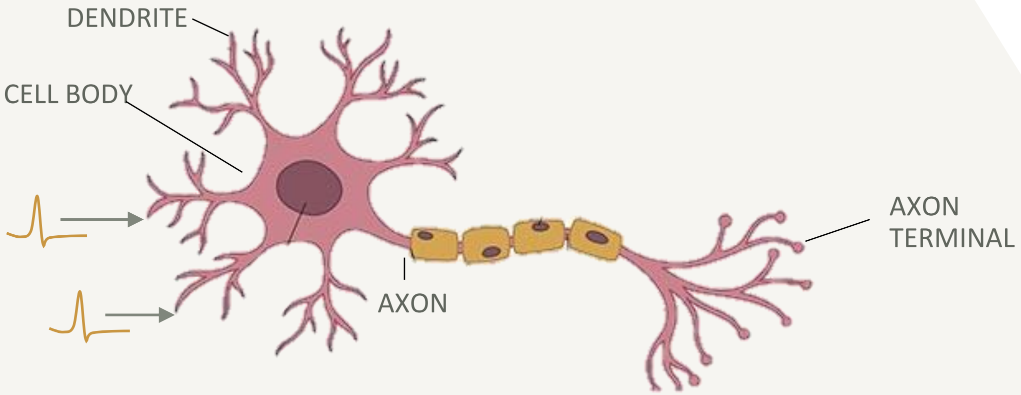
NEURAL NETWORK
A NEURON



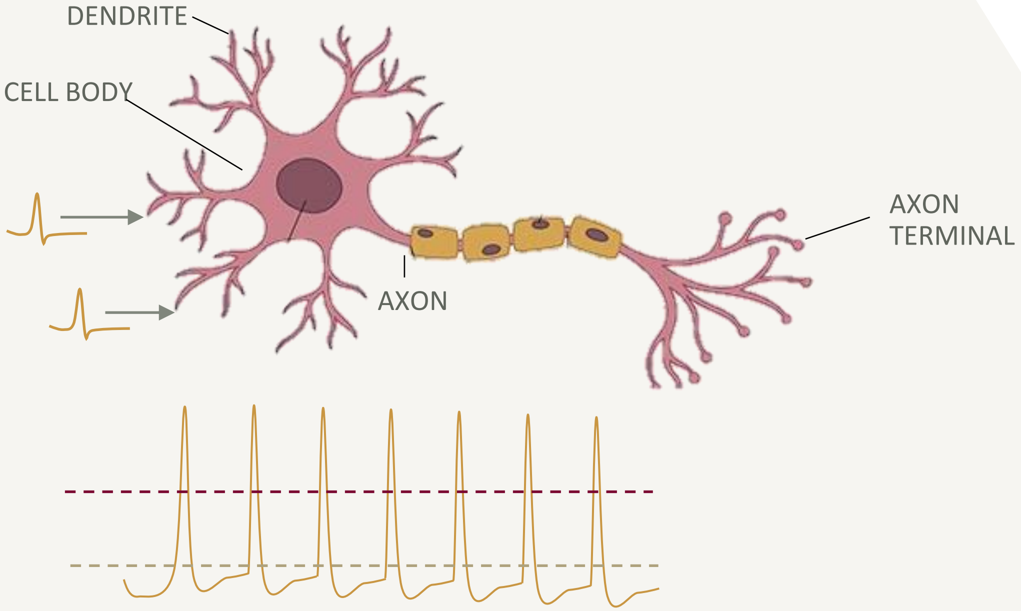
NEURAL NETWORK
A NEURON



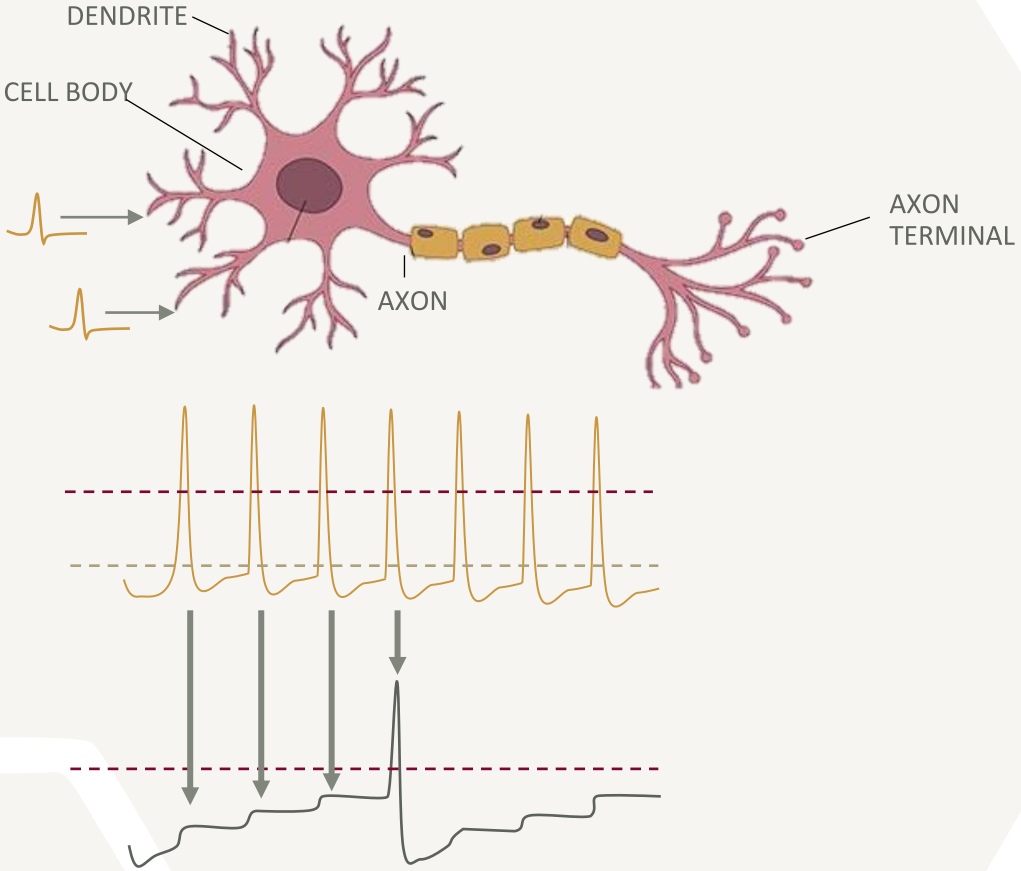
NEURAL NETWORK
A NEURON



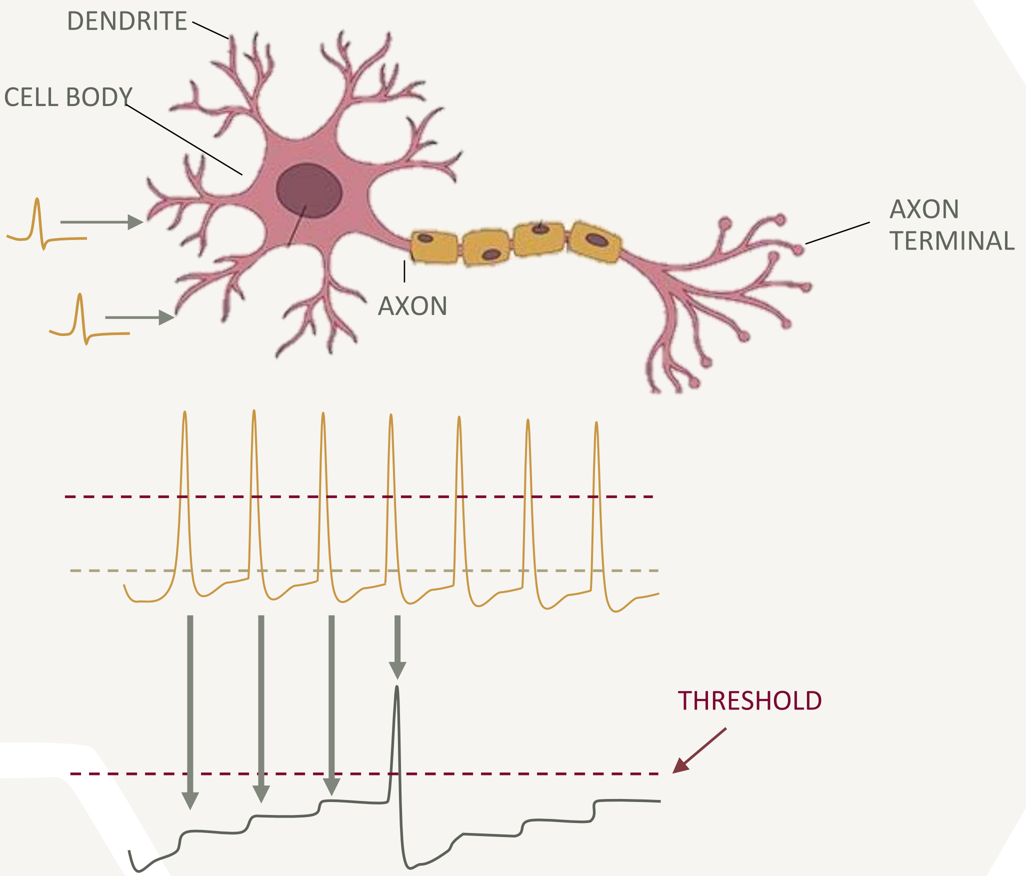
NEURAL NETWORK
A NEURON



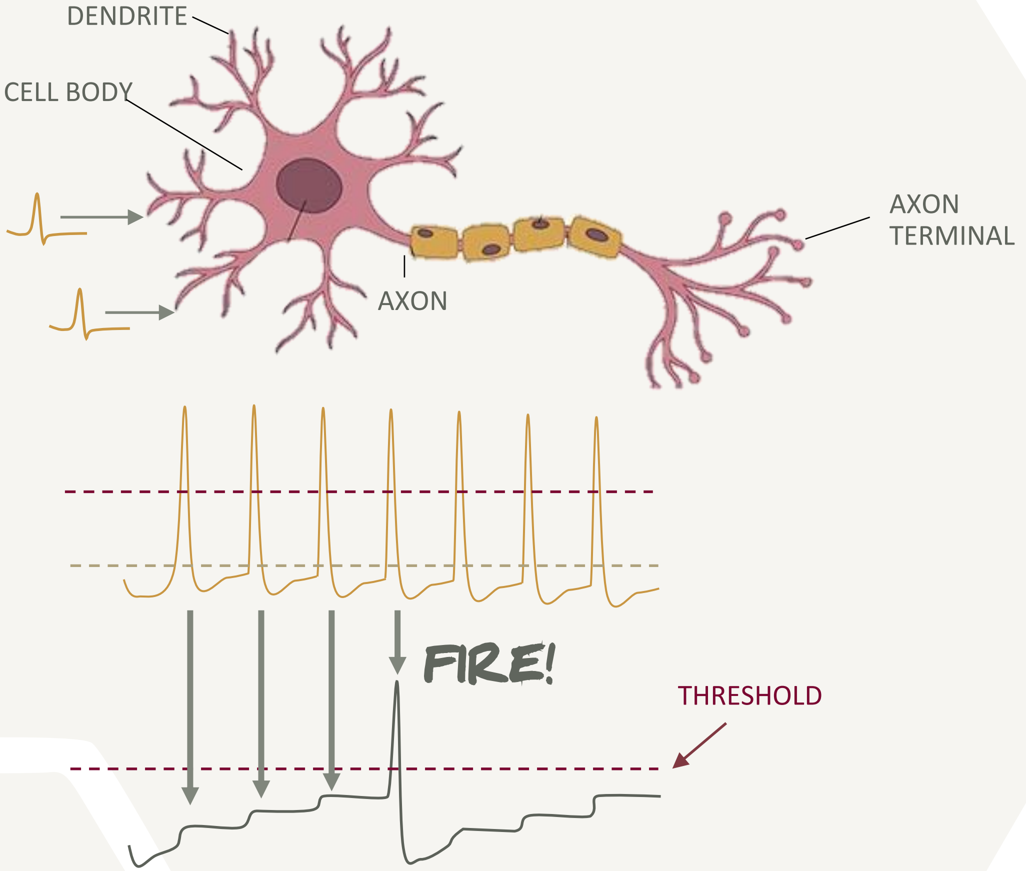
NEURAL NETWORK
A NEURON



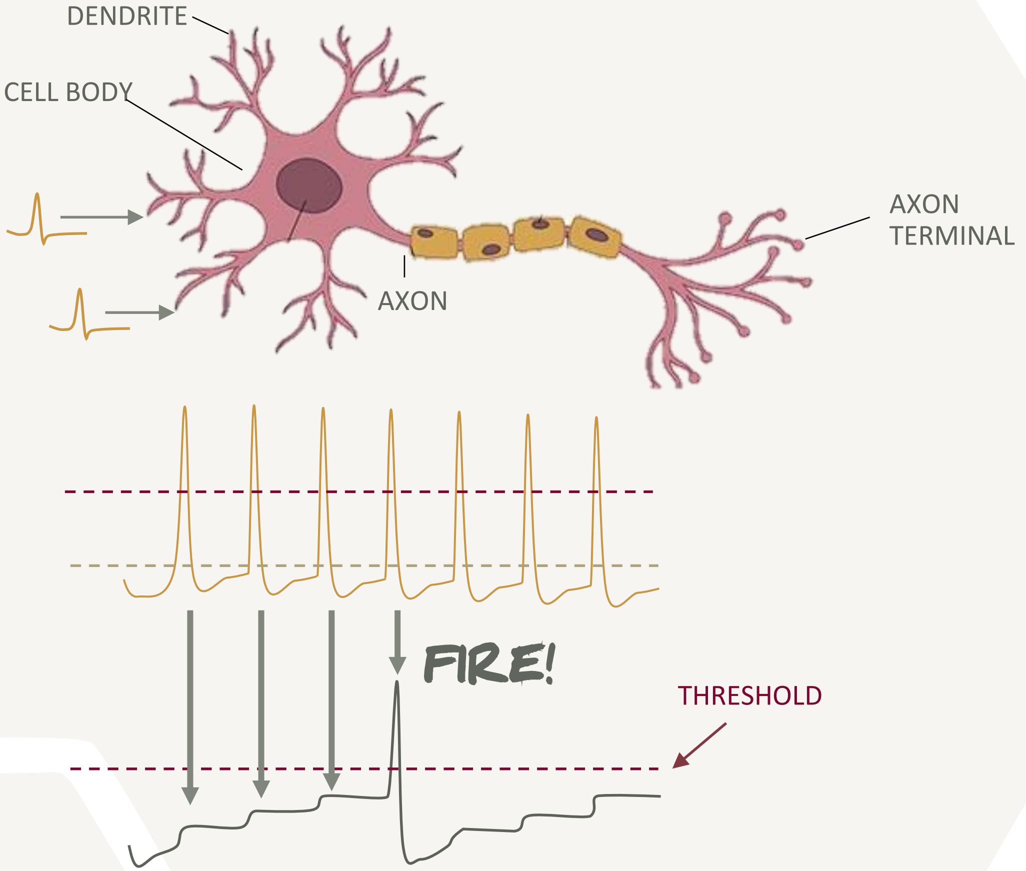
NEURAL NETWORK
A NEURON



NEURAL NETWORK
A NEURON

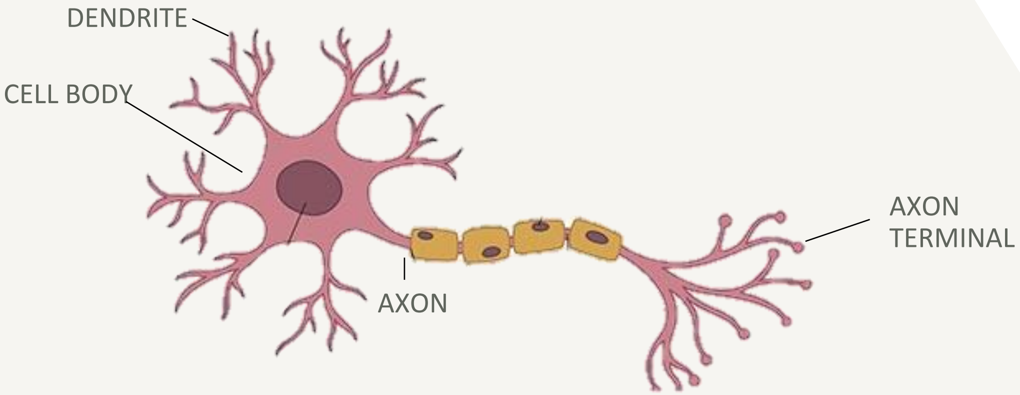


NEURAL NETWORK
A NEURON



Comic by Johannes Kretzschmar

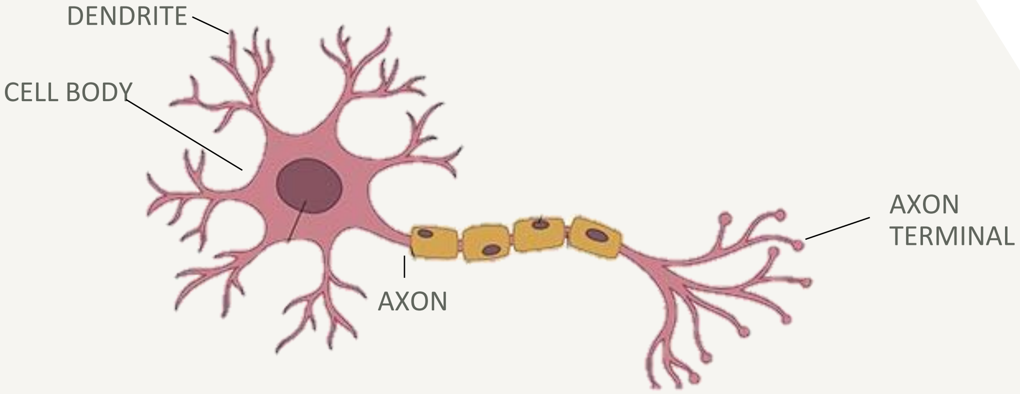
NEURAL NETWORK
A NEURON



INPUT →

Comic by Johannes Kretzschmar

NEURAL NETWORK
A NEURON

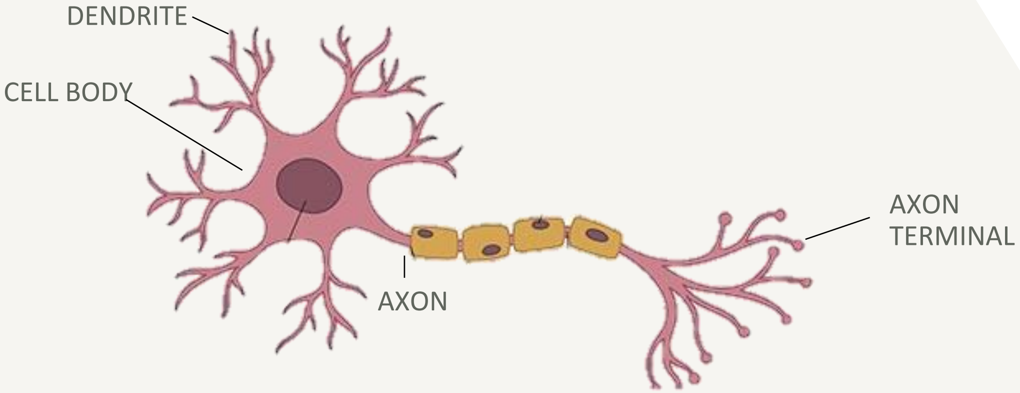


INPUT → → OUTPUT

Comic by Johannes Kretzschmar

NEURAL NETWORK

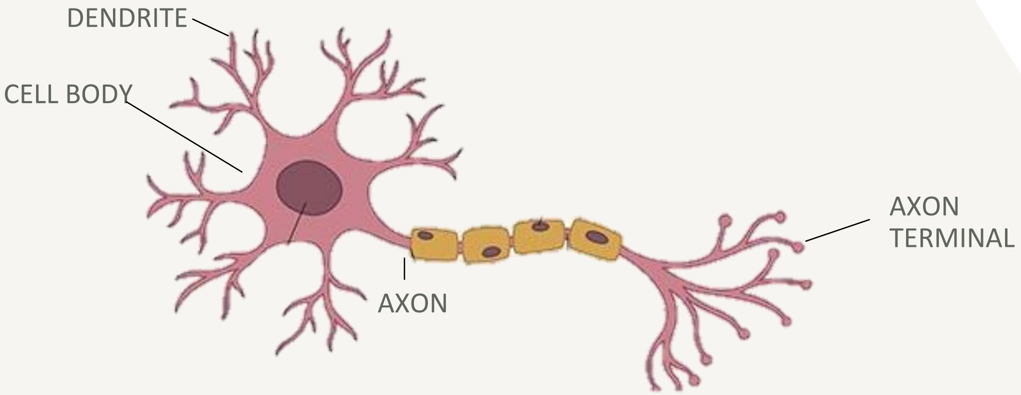
A NEURON



INPUT → TRANS-FORMATION → OUTPUT

Comic by Johannes Kretzschmar

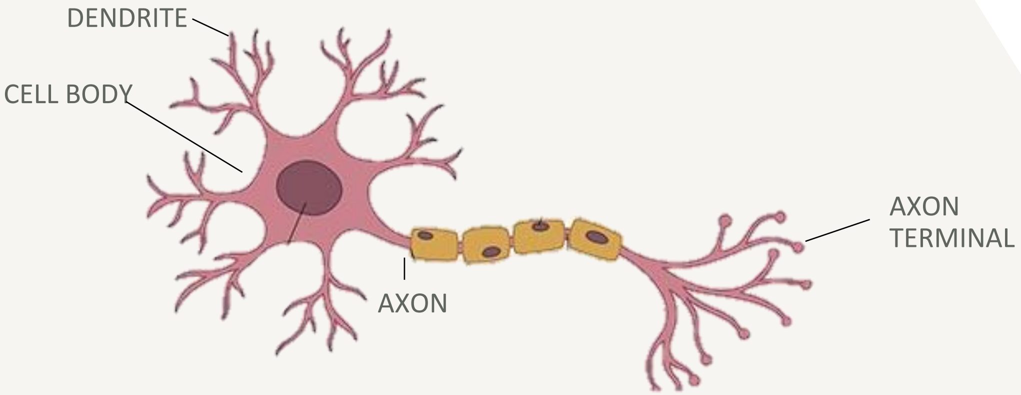
NEURAL NETWORK
A NEURON



$x \in X$ → TRANS-FORMATION → OUTPUT

Comic by Johannes Kretzschmar

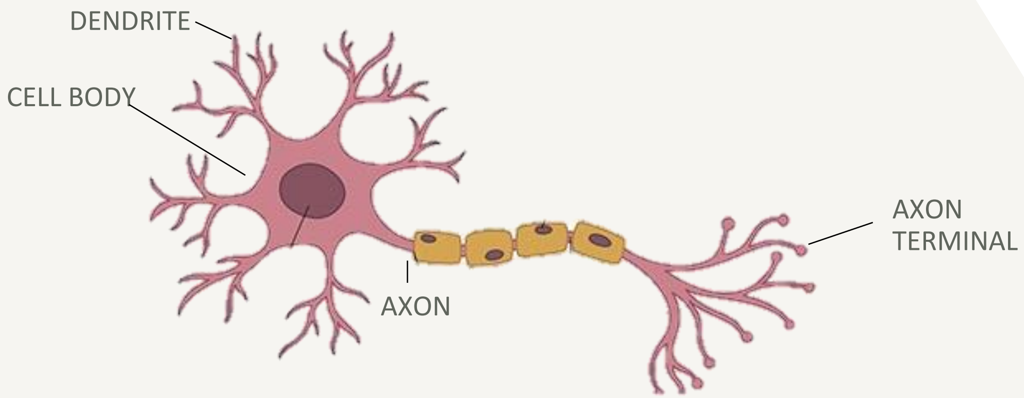
NEURAL NETWORK
A NEURON



$x \in X$ → TRANS-FORMATION → $\hat{y}$

Comic by Johannes Kretzschmar

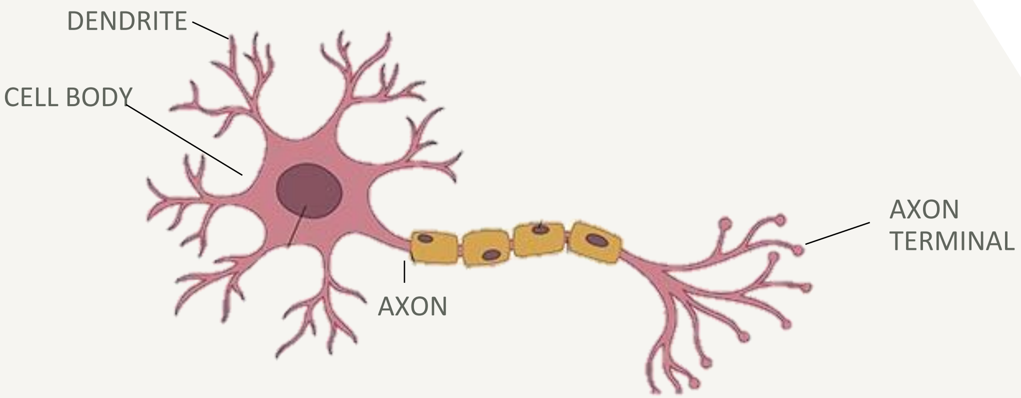
NEURAL NETWORK
A NEURON



$$x \in X \longrightarrow f(x) \longrightarrow \hat{y}$$

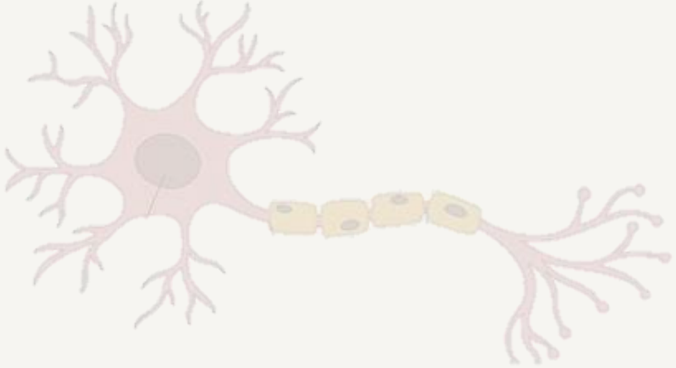
Comic by Johannes Kretzschmar

NEURAL NETWORK
A PERCEPTRON



$$x \in X \longrightarrow f(x) \longrightarrow \hat{y}$$

NEURAL NETWORK
A PERCEPTRON

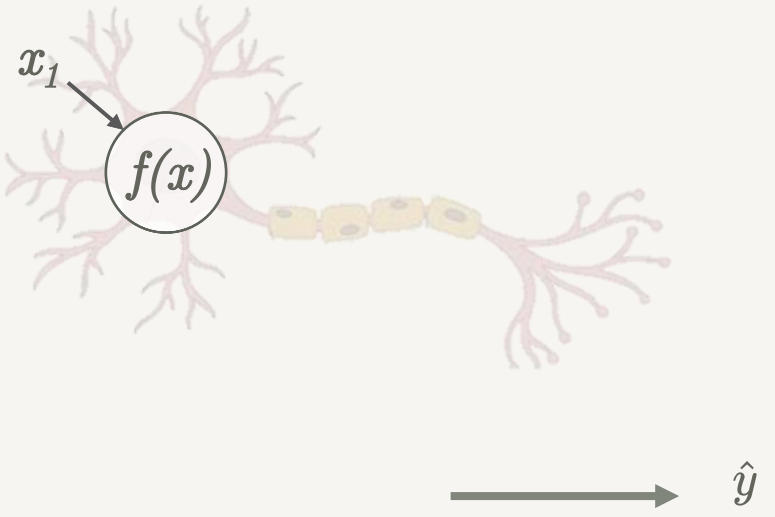


$$x \in X \longrightarrow f(x) \longrightarrow \hat{y}$$

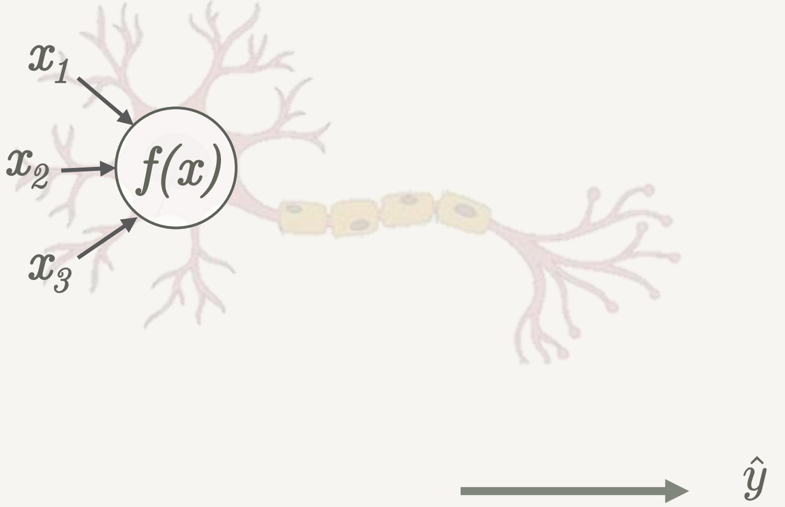
NEURAL NETWORK
A PERCEPTRON

$$x \in X \longrightarrow \qquad\qquad \longrightarrow \hat{y}$$

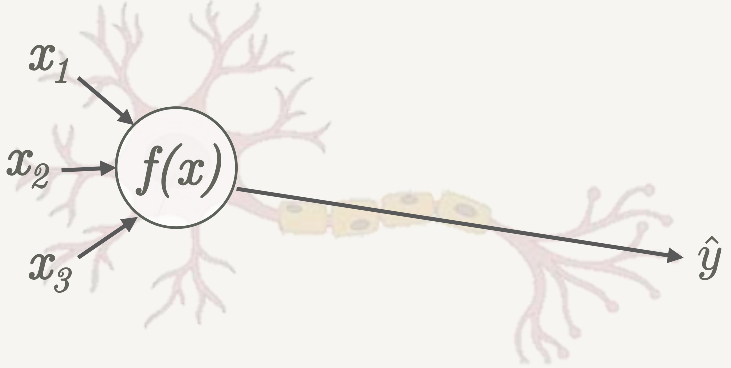
NEURAL NETWORK
A PERCEPTRON



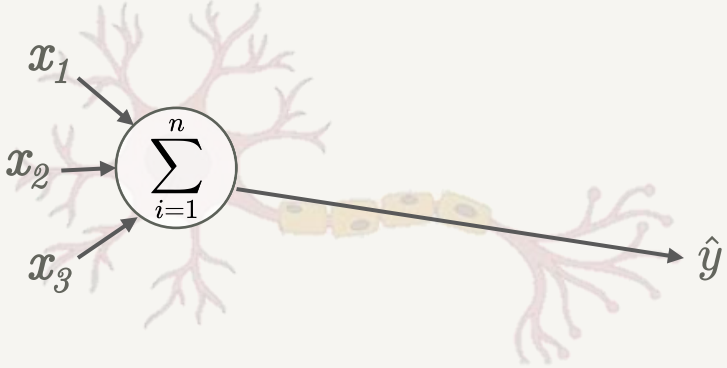
NEURAL NETWORK
A PERCEPTRON



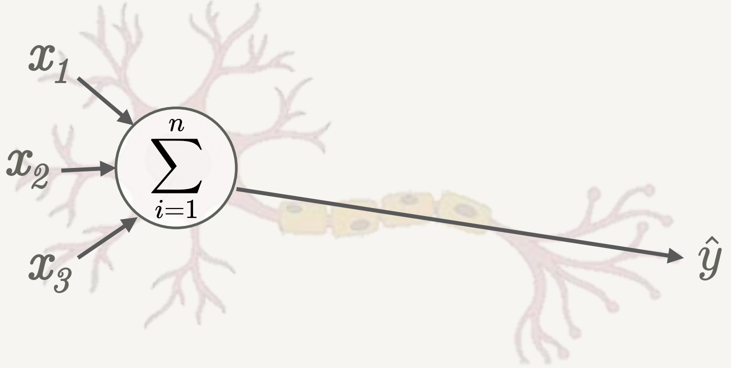
NEURAL NETWORK
A PERCEPTRON

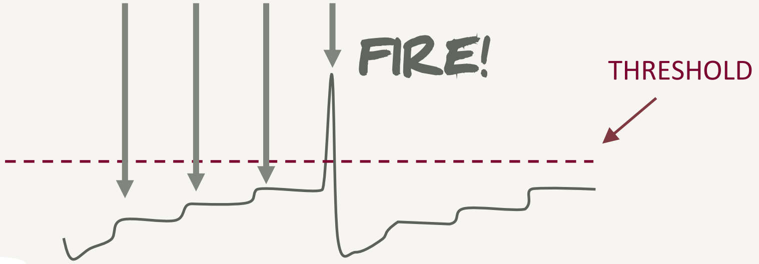


NEURAL NETWORK
A PERCEPTRON

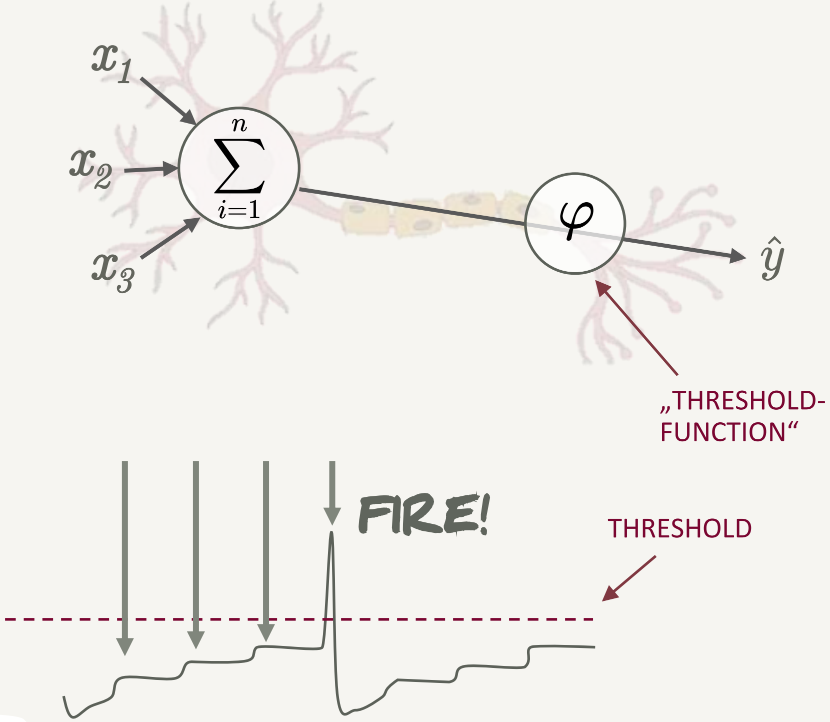


NEURAL NETWORK
A PERCEPTRON





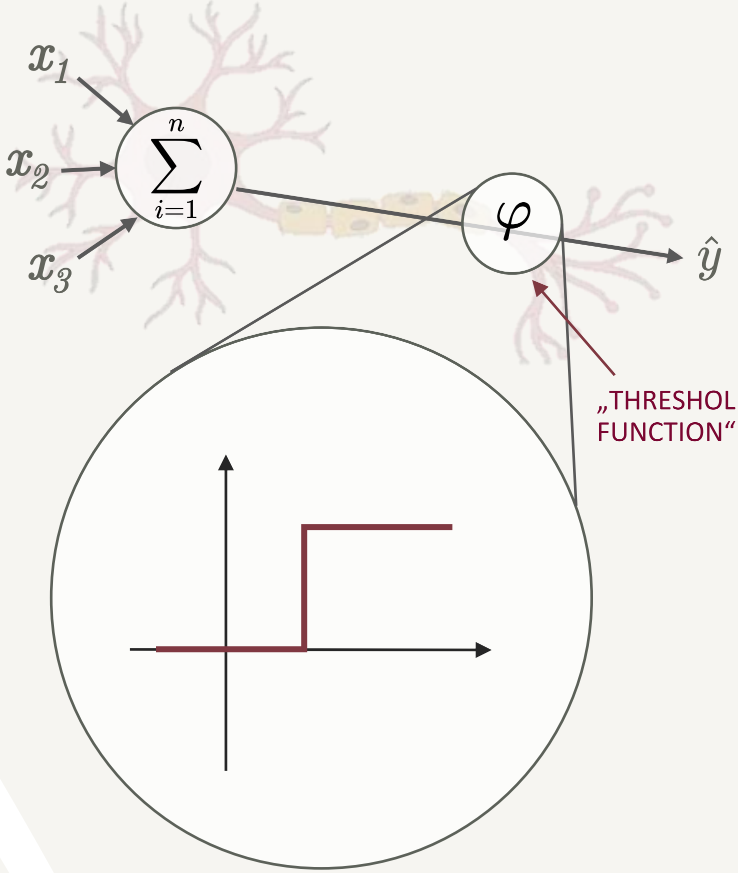
NEURAL NETWORK
A PERCEPTRON



„THRESHOLD-FUNCTION“

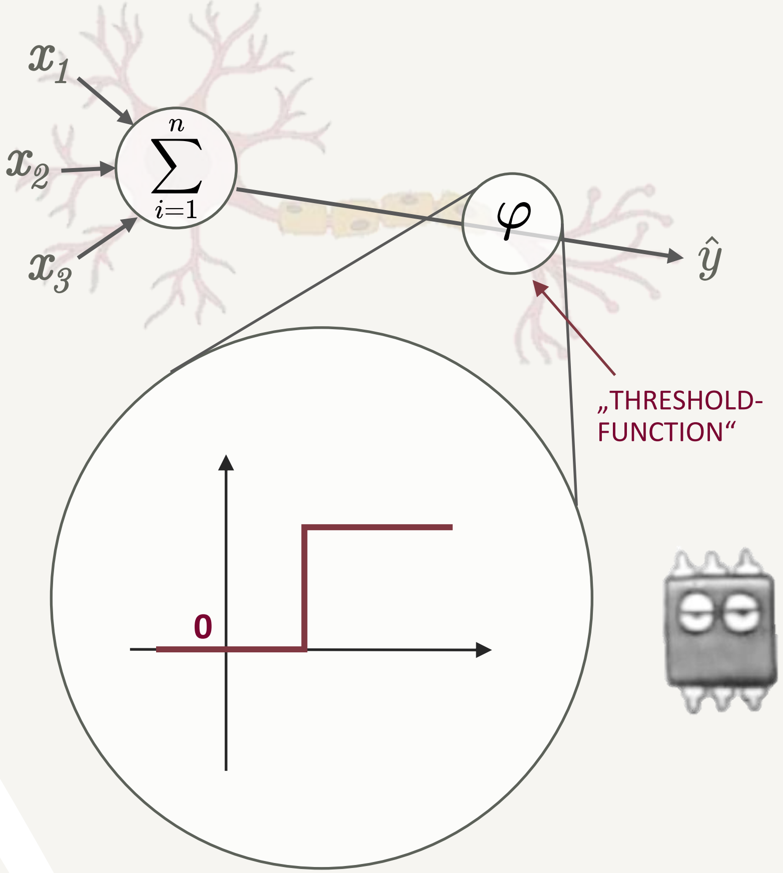
THRESHOLD

NEURAL NETWORK
A PERCEPTRON



„THRESHOLD-FUNCTION“

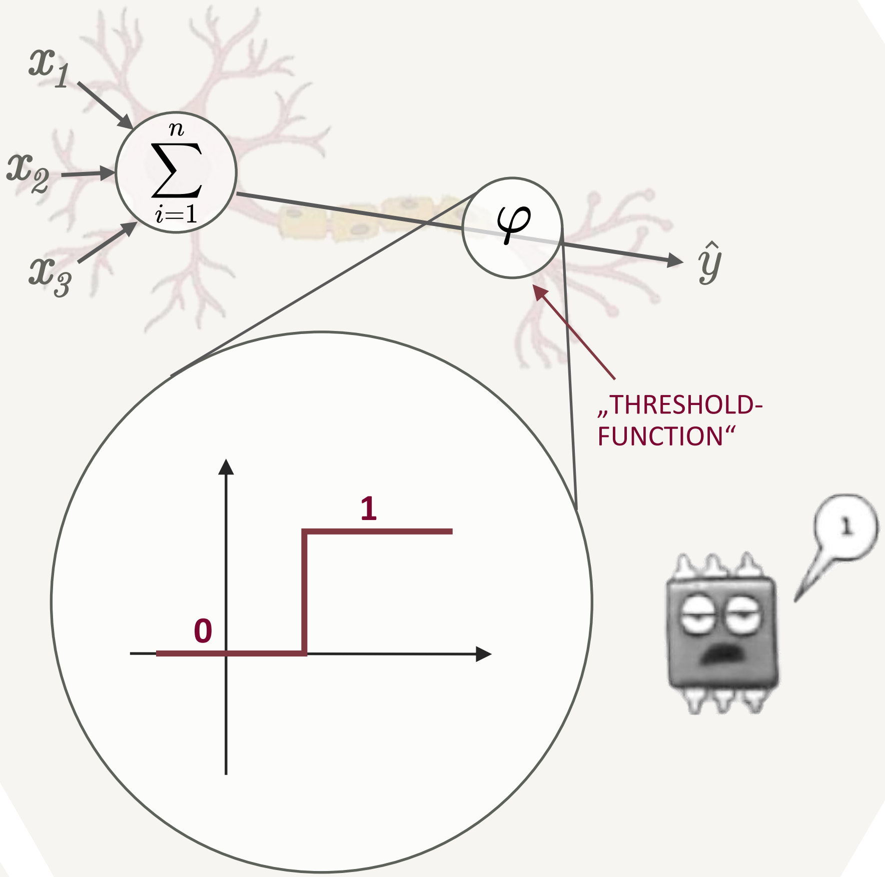
NEURAL NETWORK
A PERCEPTRON



„THRESHOLD-FUNCTION“

Comic by Johannes Kretzschmar

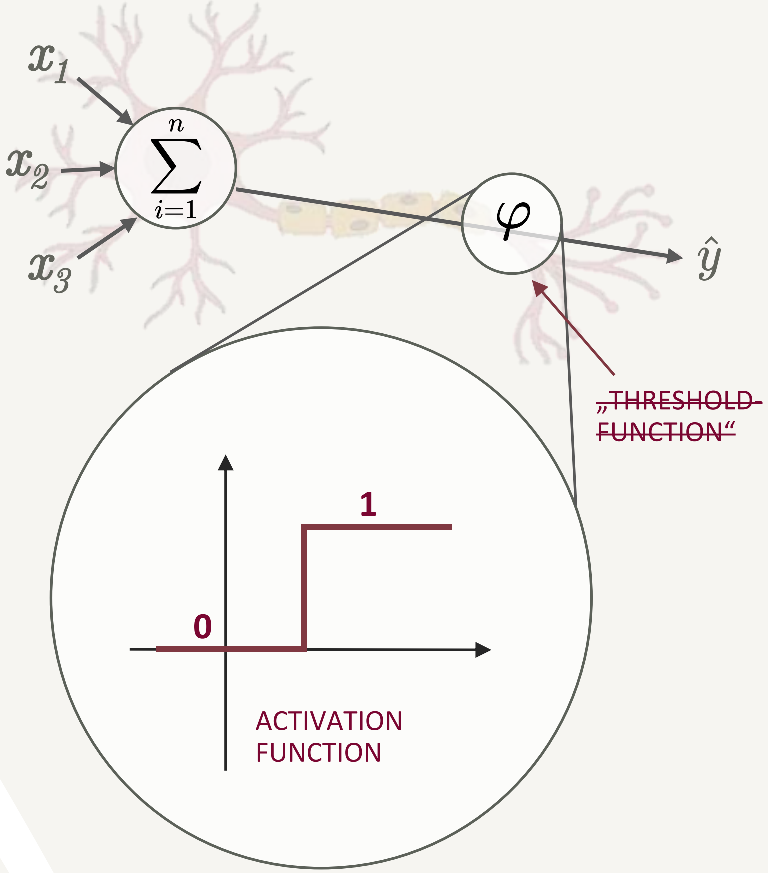
NEURAL NETWORK
A PERCEPTRON



„THRESHOLD-FUNCTION“

Comic by Johannes Kretzschmar

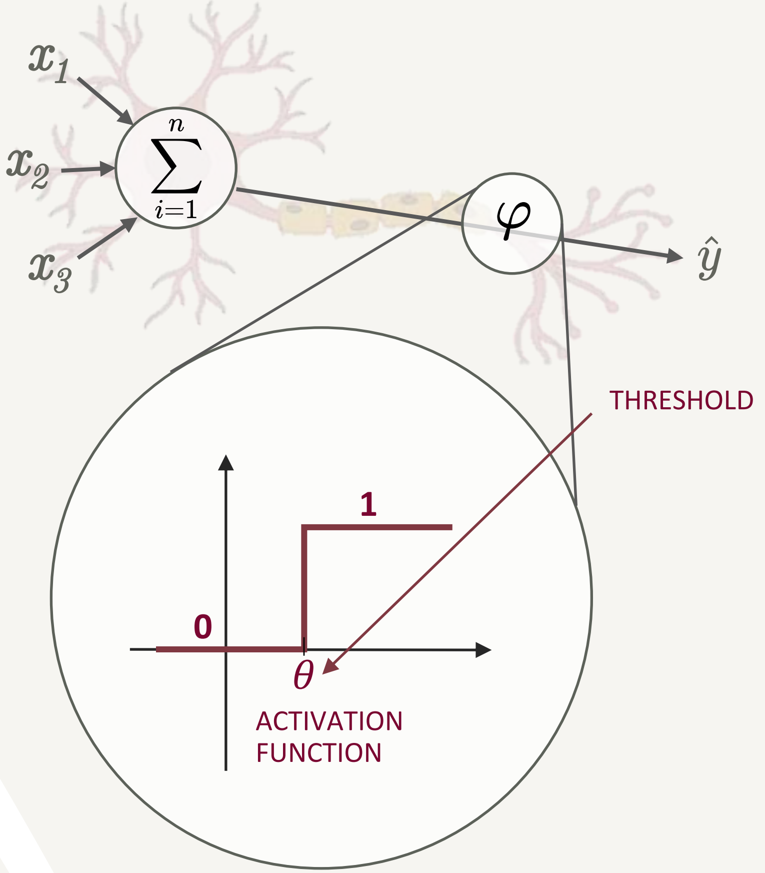
NEURAL NETWORK
A PERCEPTRON



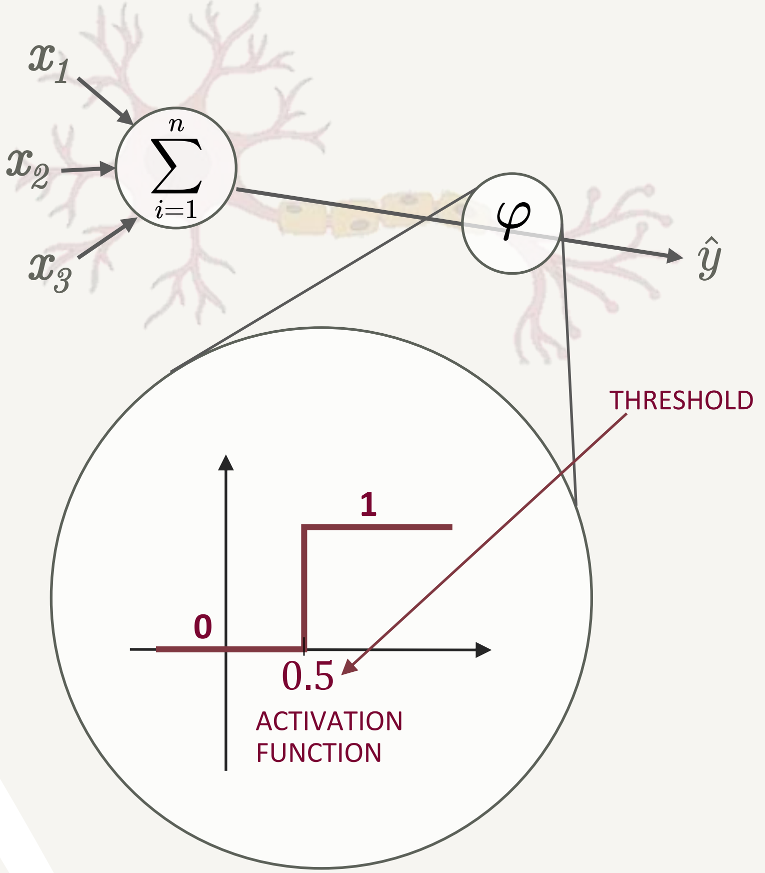
~~„THRESHOLD FUNCTION“~~

ACTIVATION FUNCTION

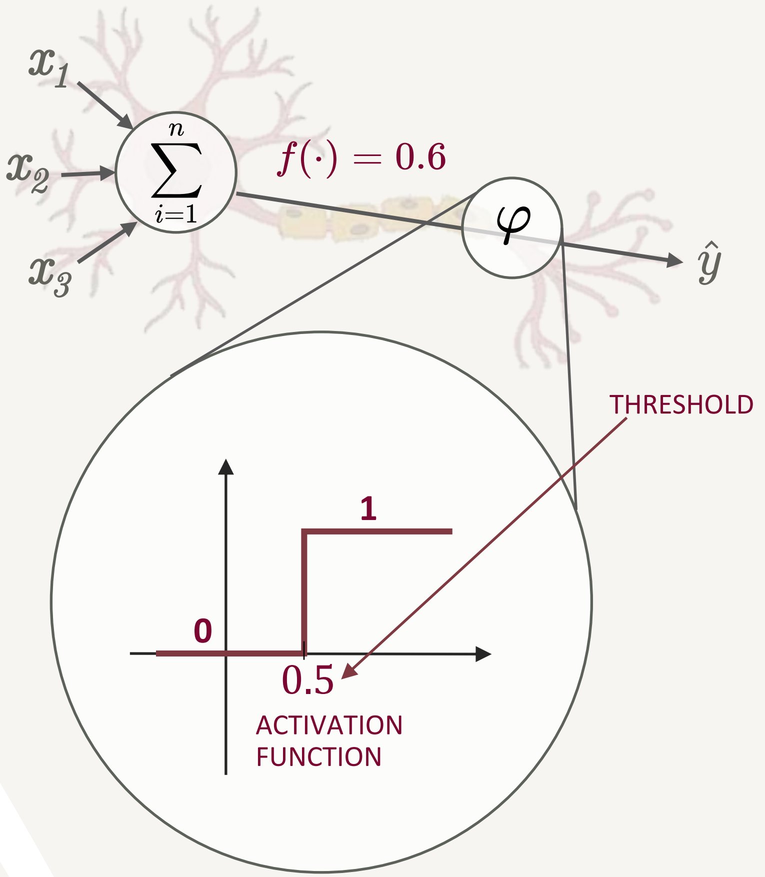
NEURAL NETWORK
A PERCEPTRON



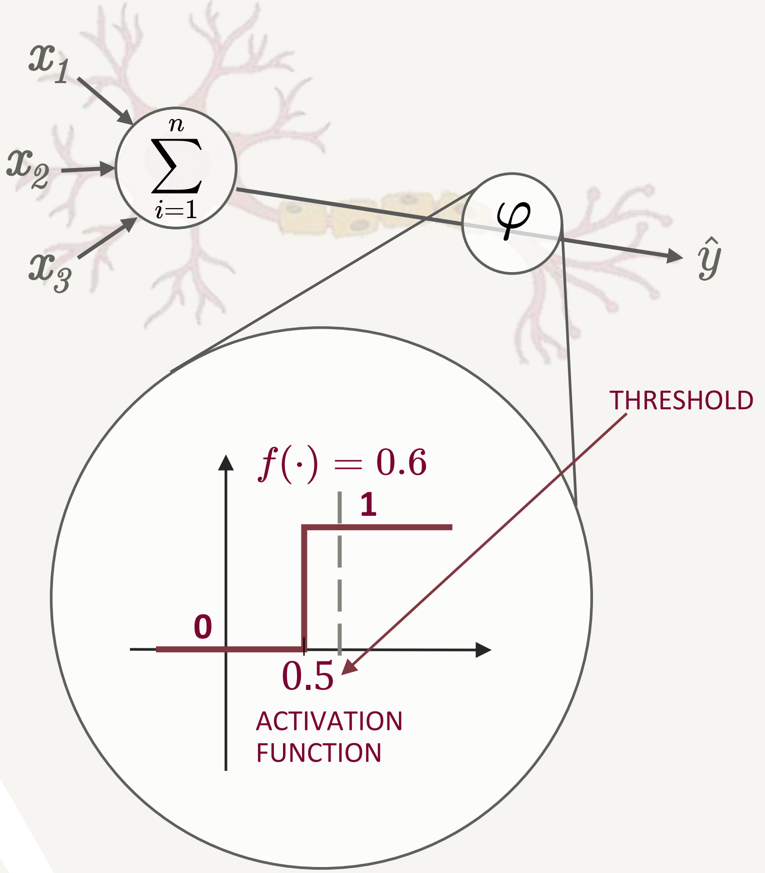
NEURAL NETWORK
A PERCEPTRON



NEURAL NETWORK
A PERCEPTRON

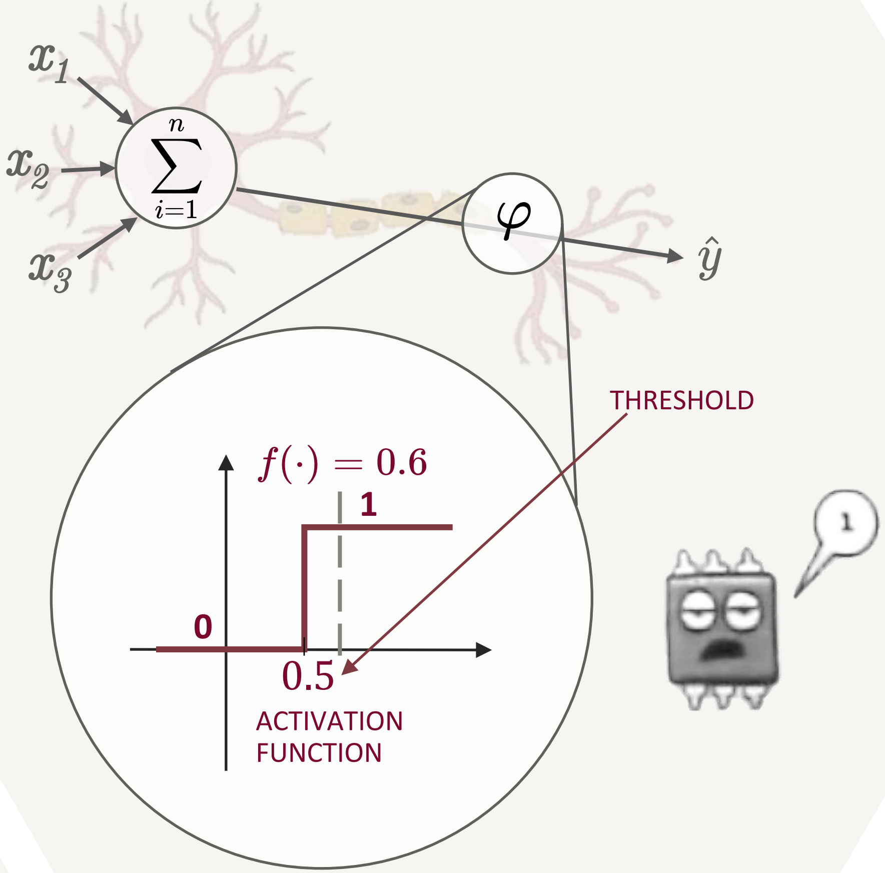


NEURAL NETWORK
A PERCEPTRON



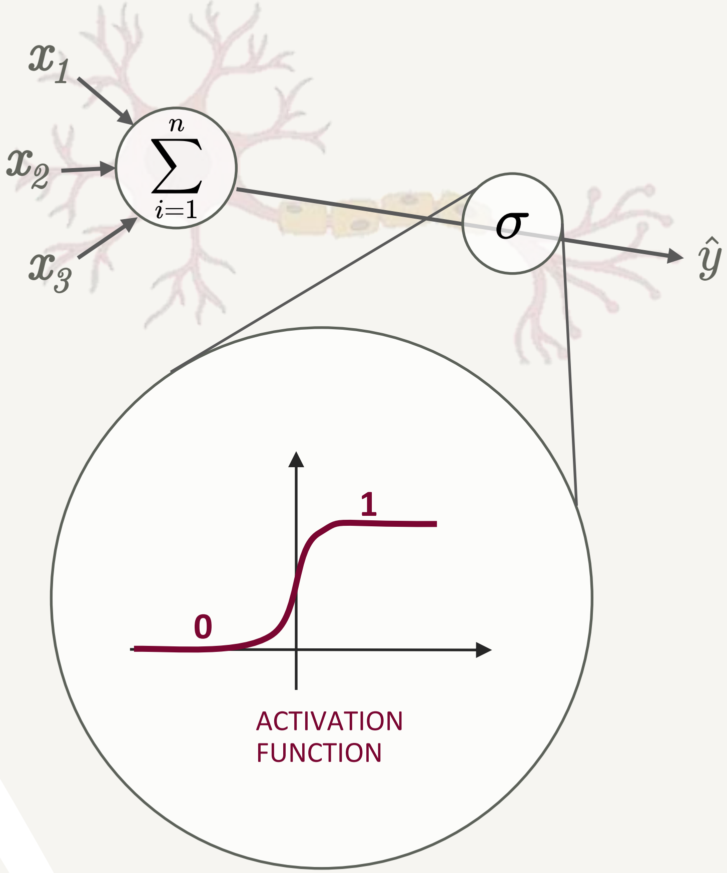
NEURAL NETWORK

A PERCEPTRON

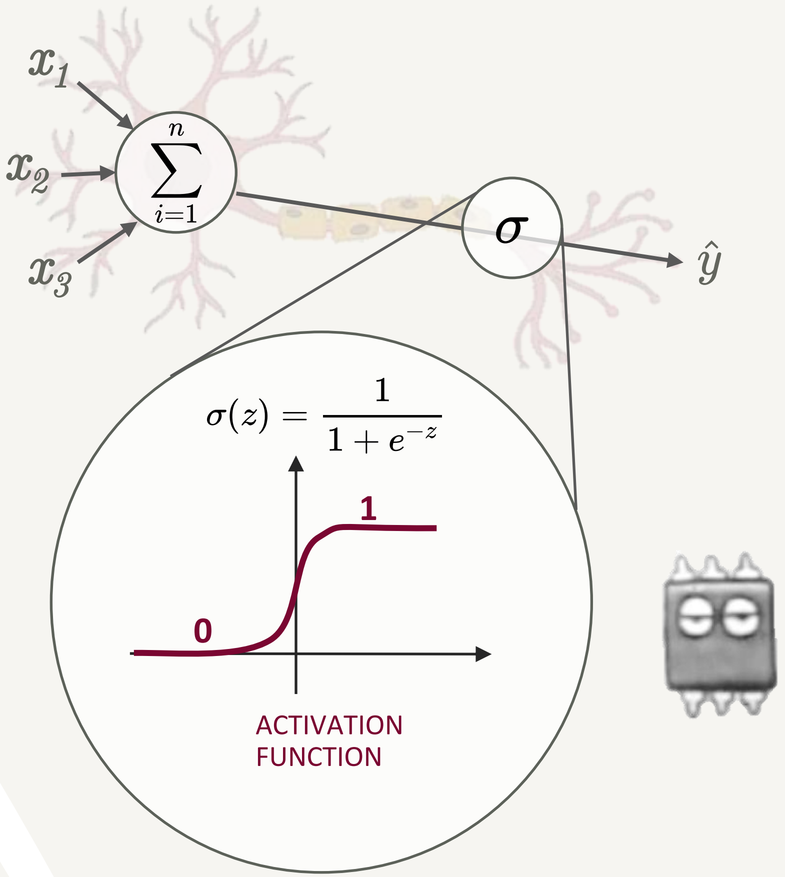


Comic by Johannes Kretzschmar

NEURAL NETWORK
A PERCEPTRON

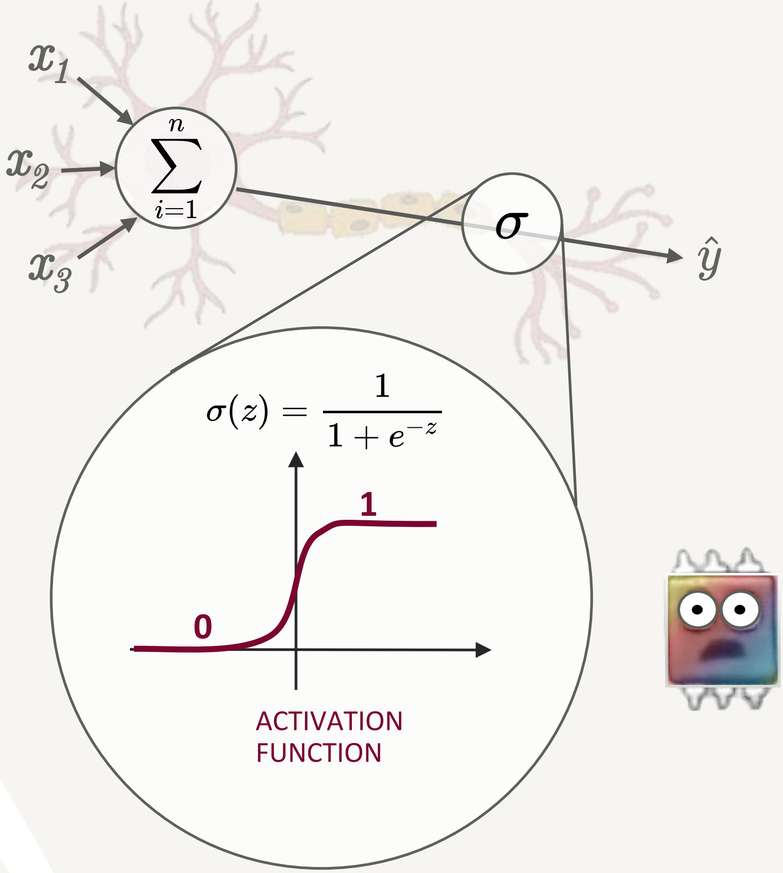


NEURAL NETWORK
A PERCEPTRON

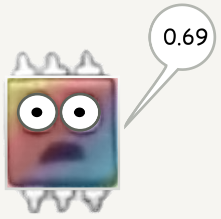


$$\sigma(z) = \frac{1}{1+e^{-z}}$$

NEURAL NETWORK
A PERCEPTRON

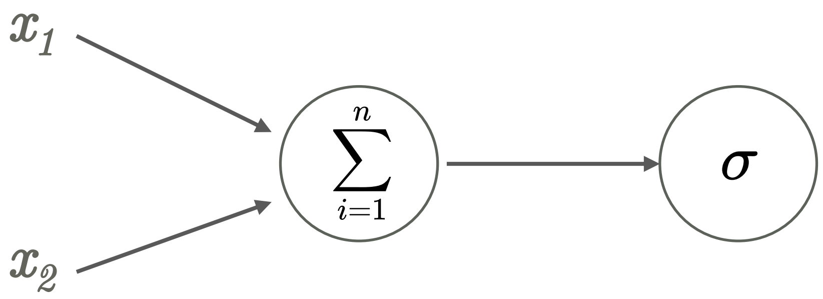


$$\sigma(z) = \frac{1}{1+e^{-z}}$$



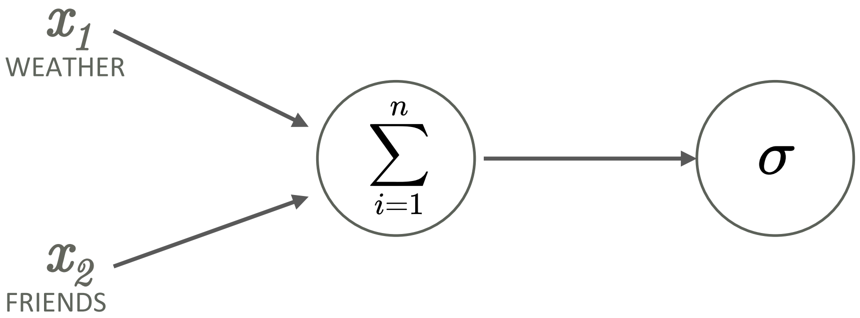
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



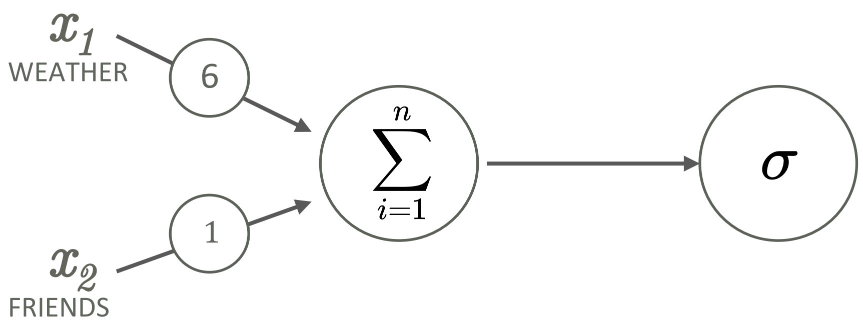
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



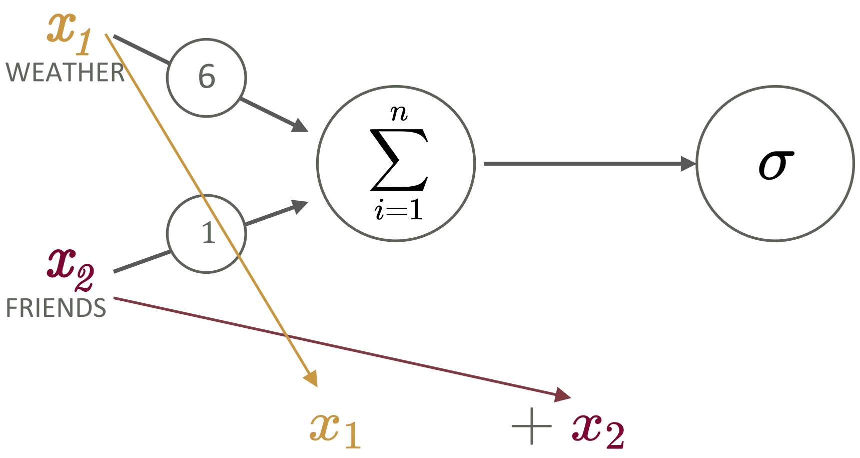
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



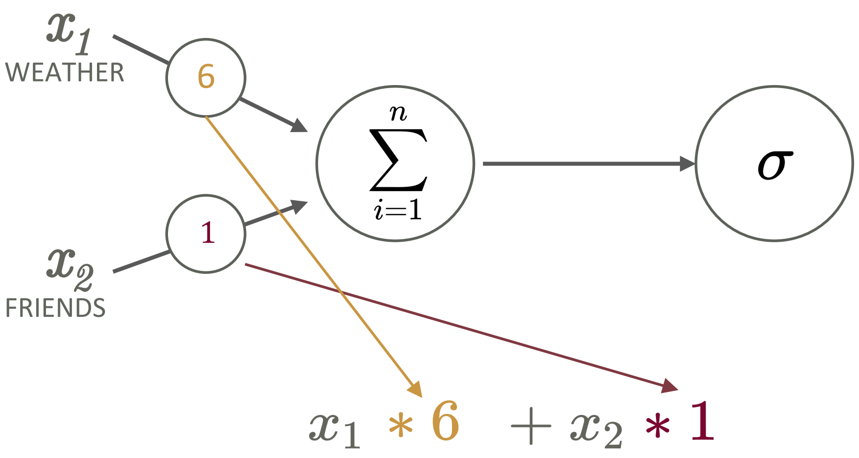
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



NEURAL NETWORK
A PERCEPTRON

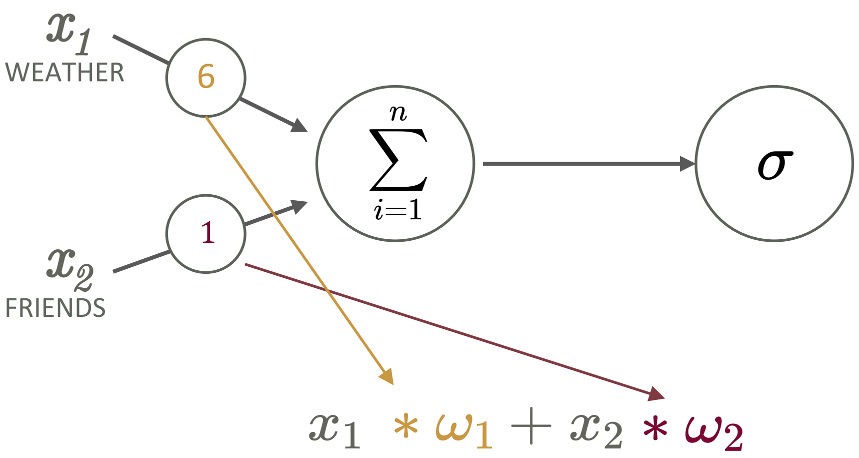
Ex: We want to decide to go **outside or not**.



$$x_1 * 6 + x_2 * 1$$

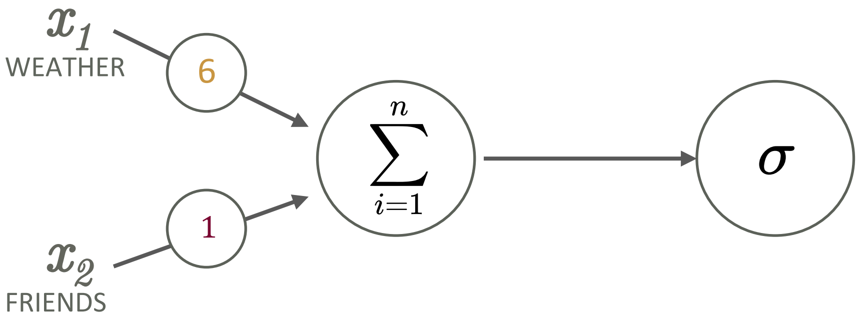
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



NEURAL NETWORK
A PERCEPTRON

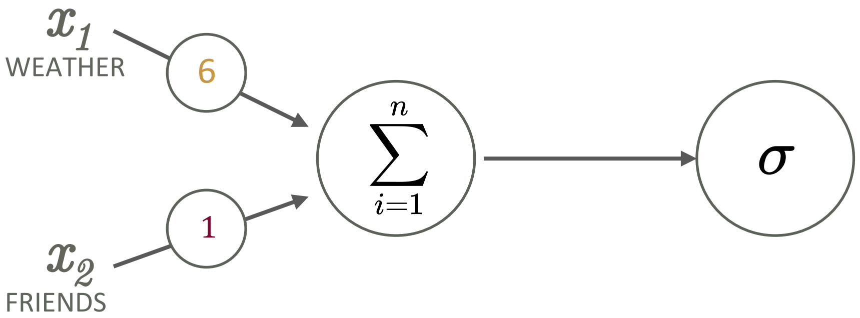
Ex: We want to decide to go **outside or not**.



$$\underbrace{x_1 * \omega_1} + \underbrace{x_2 * \omega_2}$$

NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

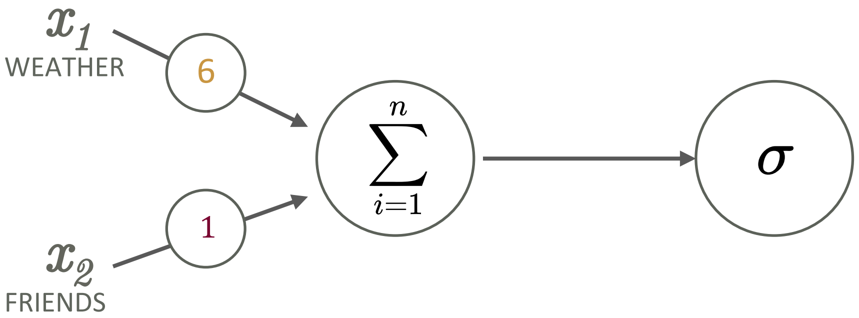


$$\underbrace{x_1 * \omega_1}_{} + \underbrace{x_2 * \omega_2}_{}$$

$$\sum_{i=1}^{n} x_i * \omega_i$$

NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



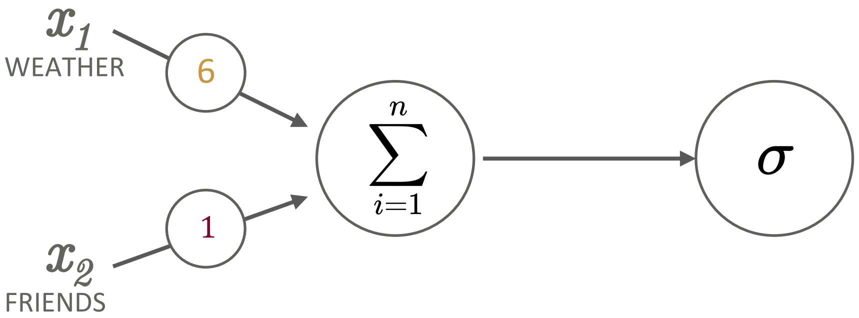
This is called the weighted sum

$$\underbrace{x_1 * \omega_1} + \underbrace{x_2 * \omega_2}$$

$$\sum_{i=1}^{n} x_i * \omega_i$$

NEURAL NETWORK
A PERCEPTRON

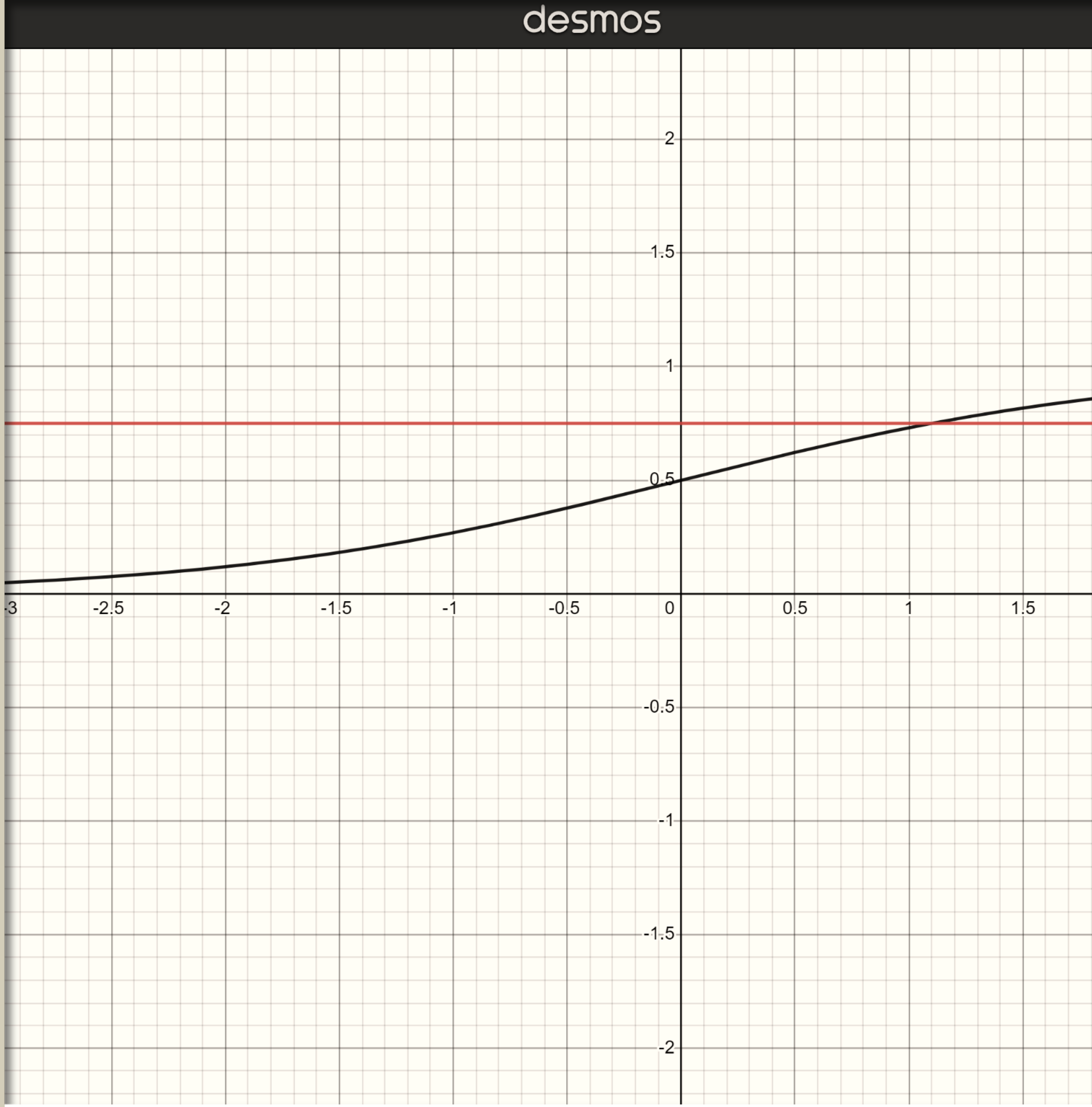
Ex: We want to decide to go **outside or not**.



$$\underbrace{x_1 * \omega_1}+\underbrace{x_2 * \omega_2}$$

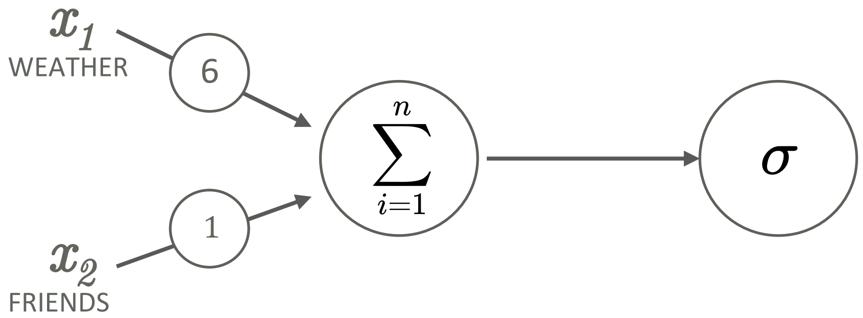
$$\sigma\left(\sum_{i=1}^{n} x_i * \omega_i\right)$$

HOW IT LOOKS LIKE…



NEURAL NETWORK
A PERCEPTRON

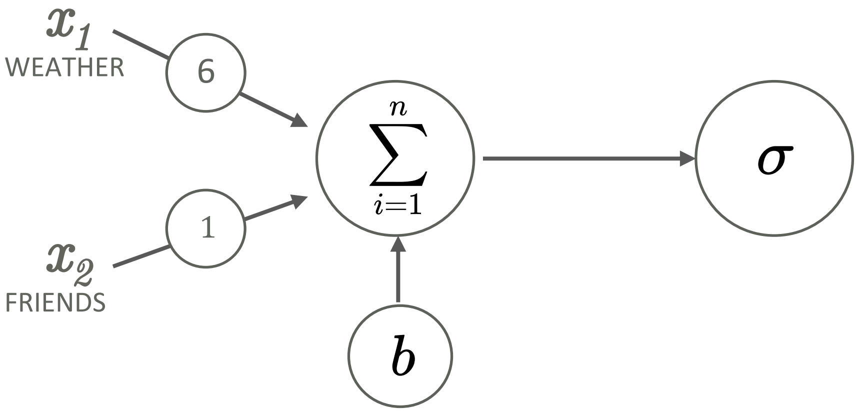
Ex: We want to decide to go **outside or not**.



$$\sigma\left(\sum_{i=1}^{n} x_i * \omega_i\right)$$

NEURAL NETWORK
A PERCEPTRON

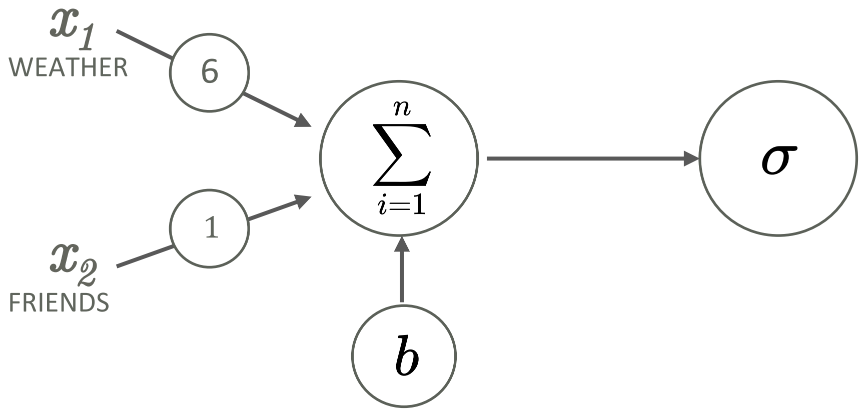
Ex: We want to decide to go **outside or not**.



$$\sigma\left(\sum_{i=1}^{n} x_i * \omega_i\right)$$

NEURAL NETWORK
A PERCEPTRON

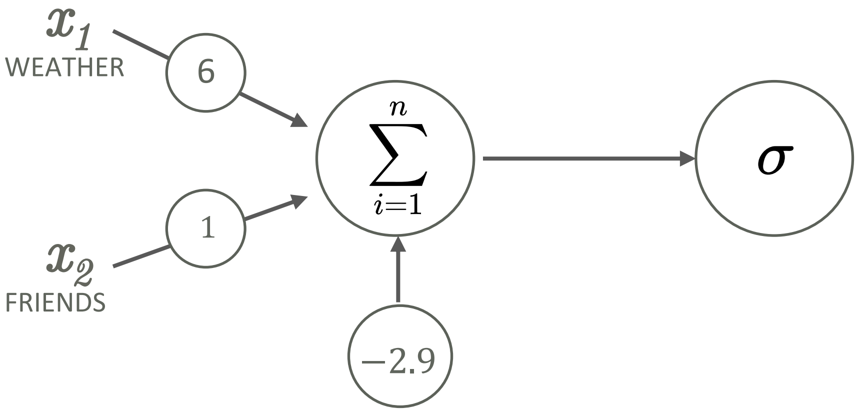
Ex: We want to decide to go **outside or not**.



$$\sigma\left(b + \sum_{i=1}^{n} x_i * \omega_i\right)$$

NEURAL NETWORK
A PERCEPTRON

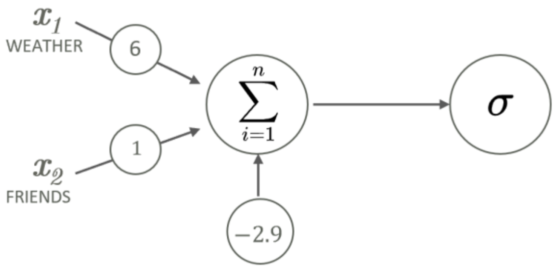
Ex: We want to decide to go **outside or not**.



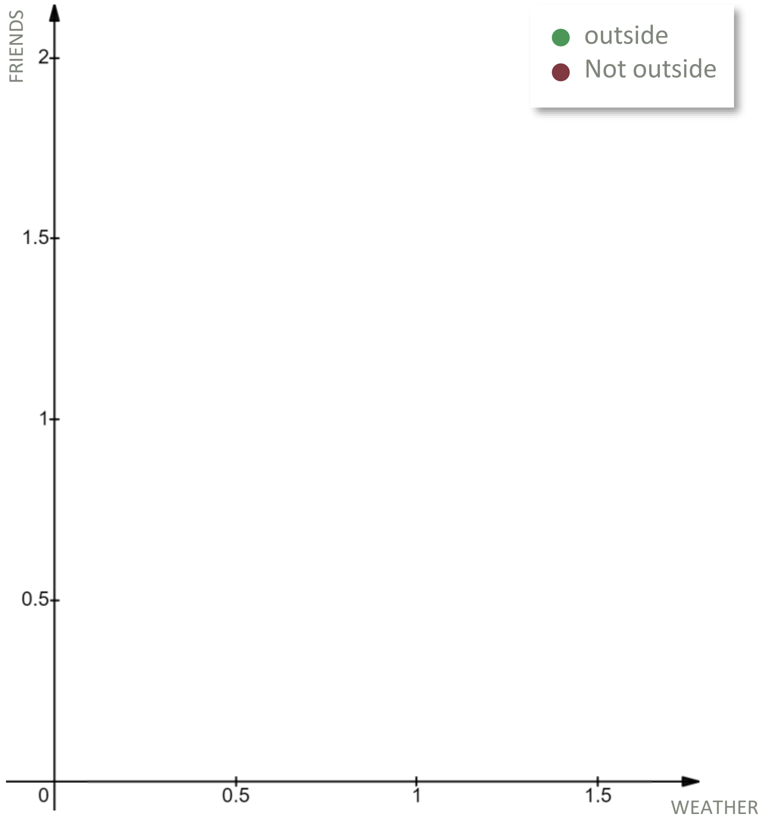
$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right)$$

NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

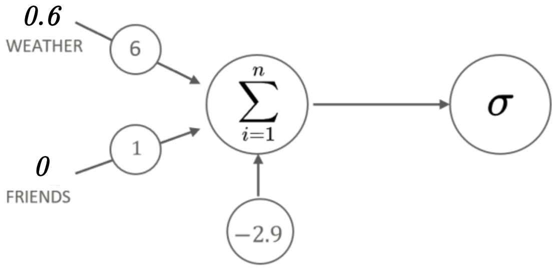


$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right)$$

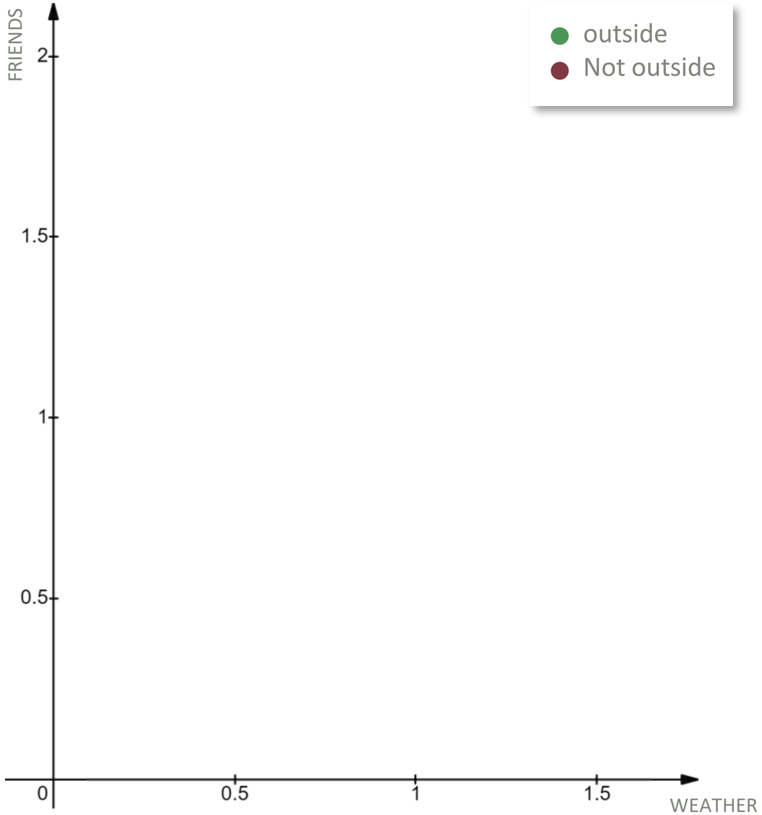


NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

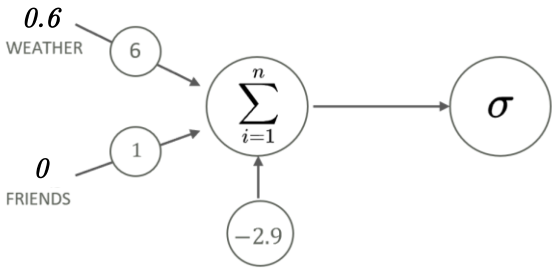


$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right)$$

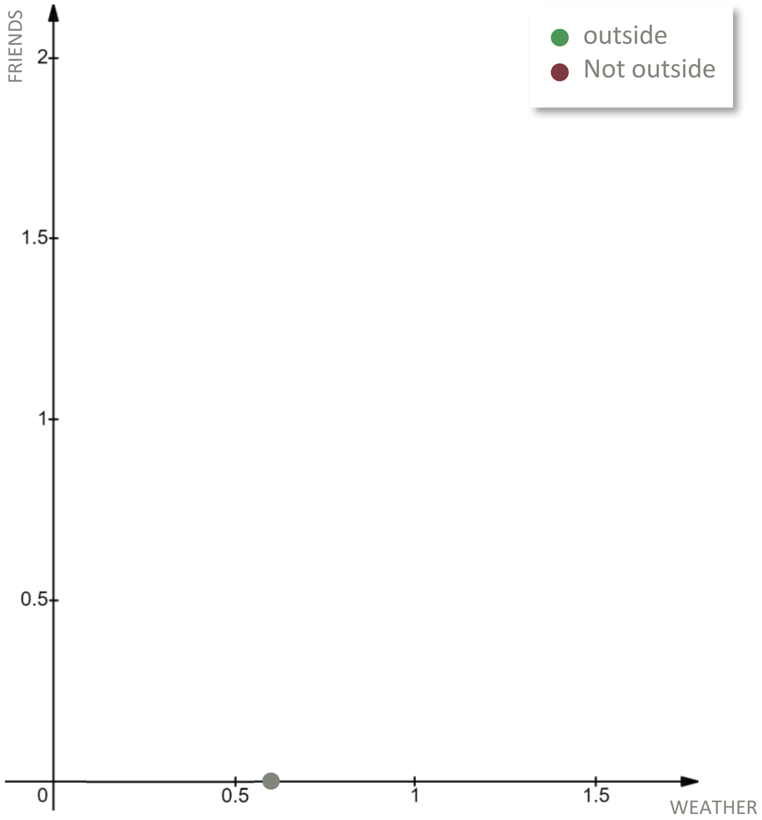


NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

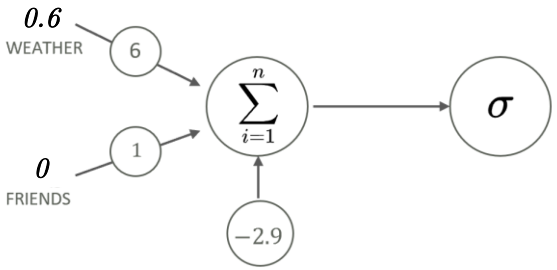


$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right)$$



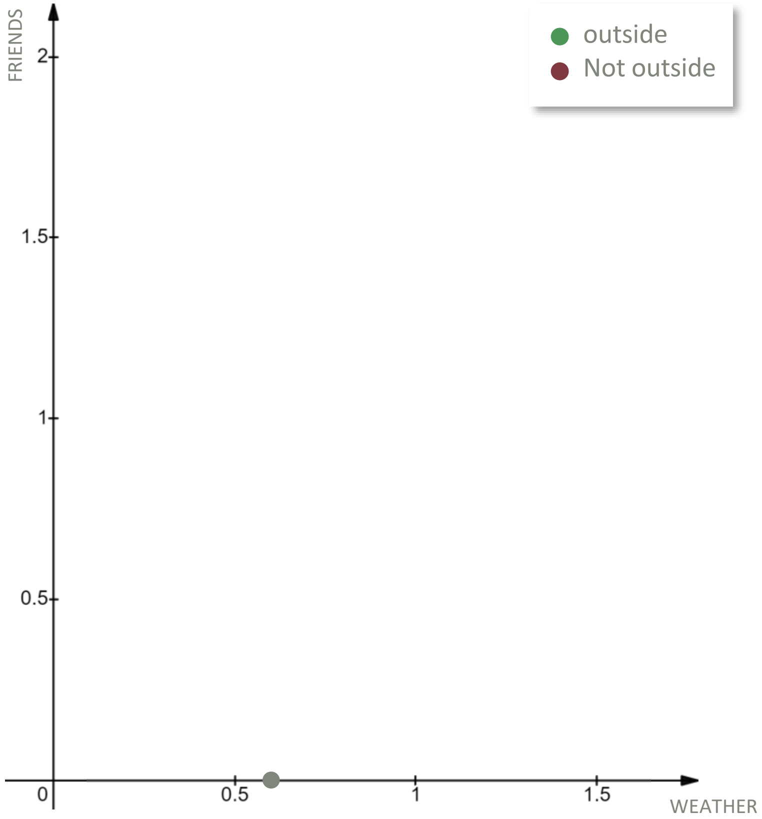
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



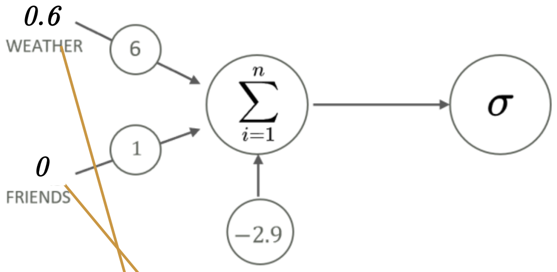
$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right)$$

$$\sigma\left(-2.9+x_1 * \omega_1 + x_2 \ * \omega_2\right)$$



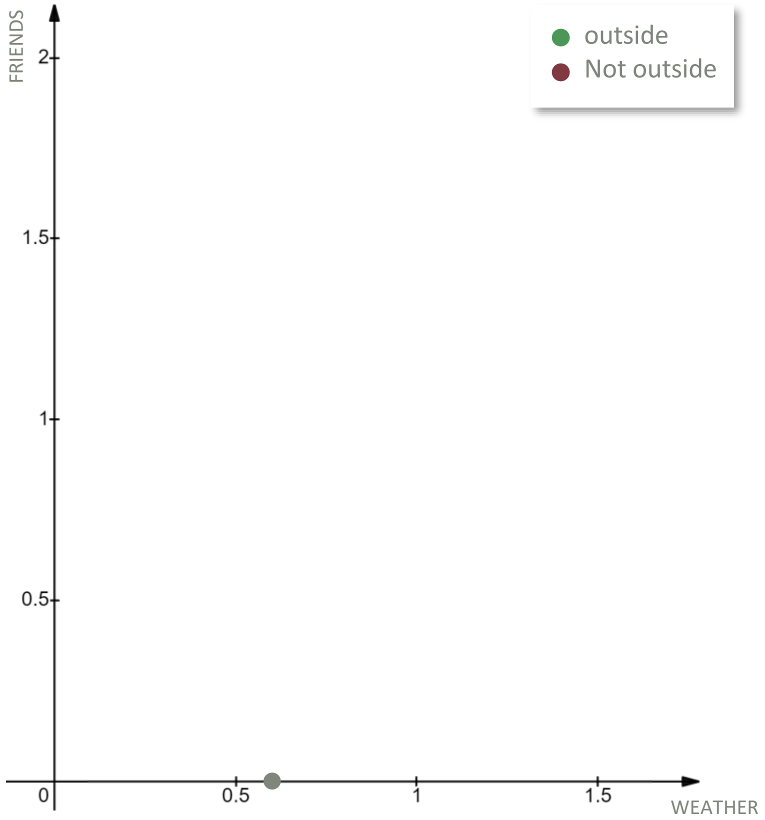
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



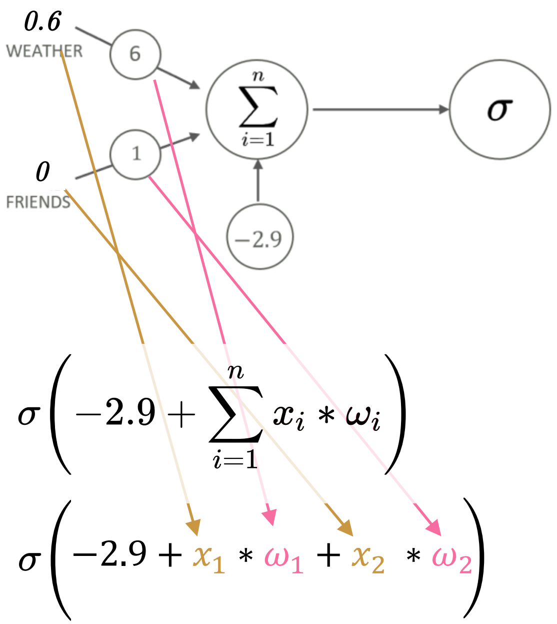
$$\sigma\left(-2.9 + \sum_{i=1}^{n} x_i * \omega_i\right)$$

$$\sigma\left(-2.9 + x_1 * \omega_1 + x_2 * \omega_2\right)$$



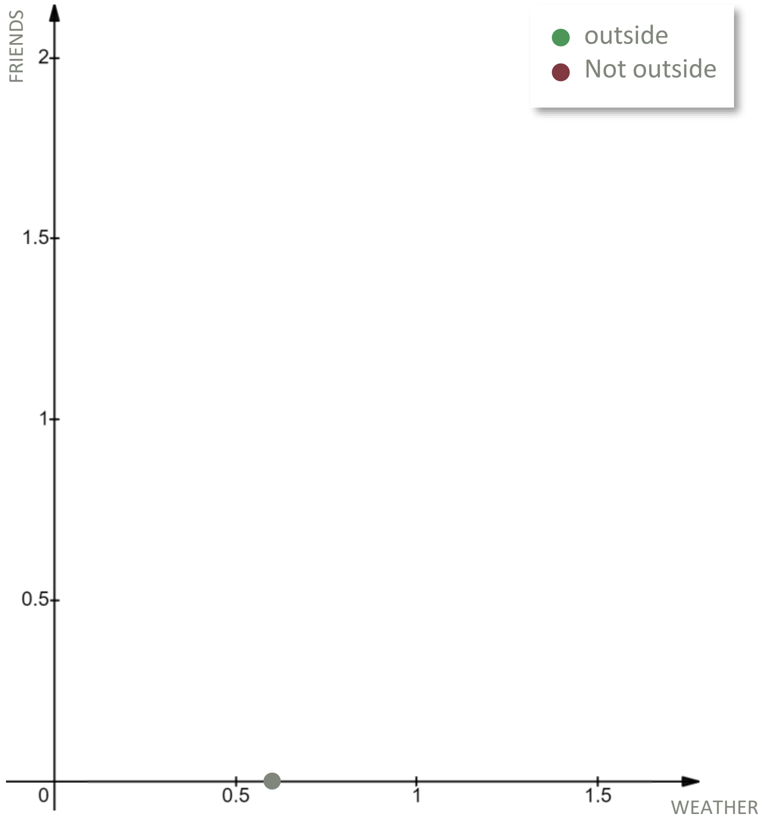
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



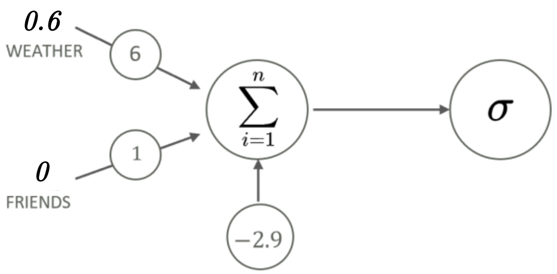
$$\sigma\left(-2.9 + \sum_{i=1}^{n} x_i * \omega_i\right)$$

$$\sigma\left(-2.9 + x_1 * \omega_1 + x_2 * \omega_2\right)$$



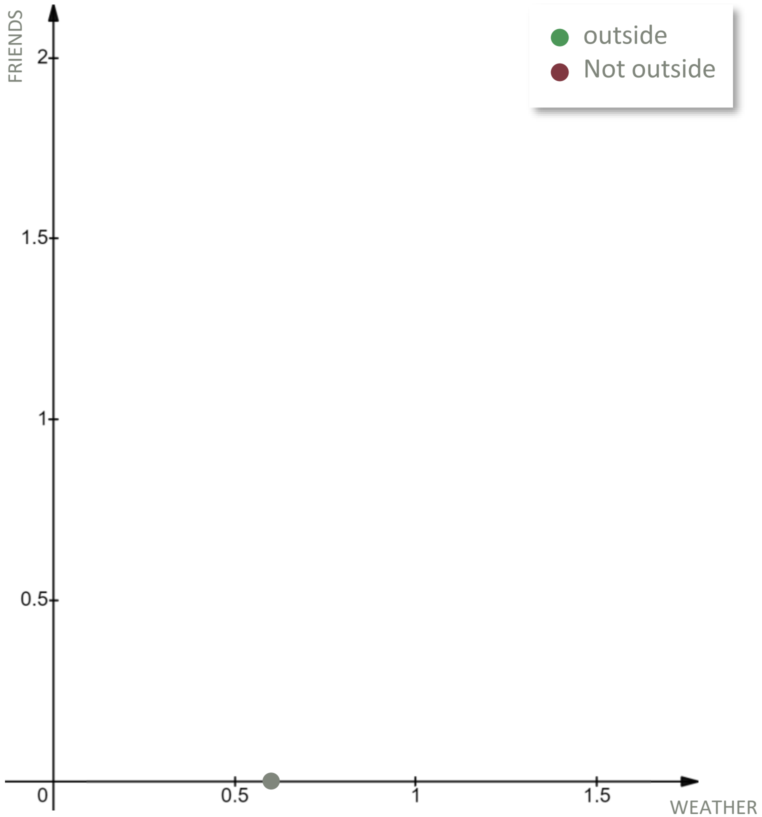
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



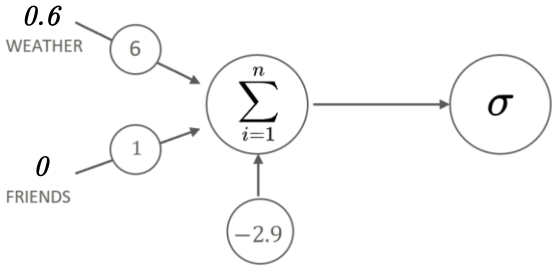
$$\sigma\left(-2.9 + \sum_{i=1}^{n} x_i * \omega_i\right)$$

$$\sigma\left(-2.9 + 0.6 * 6 \ + \ 0 \ \ * 1\right)$$



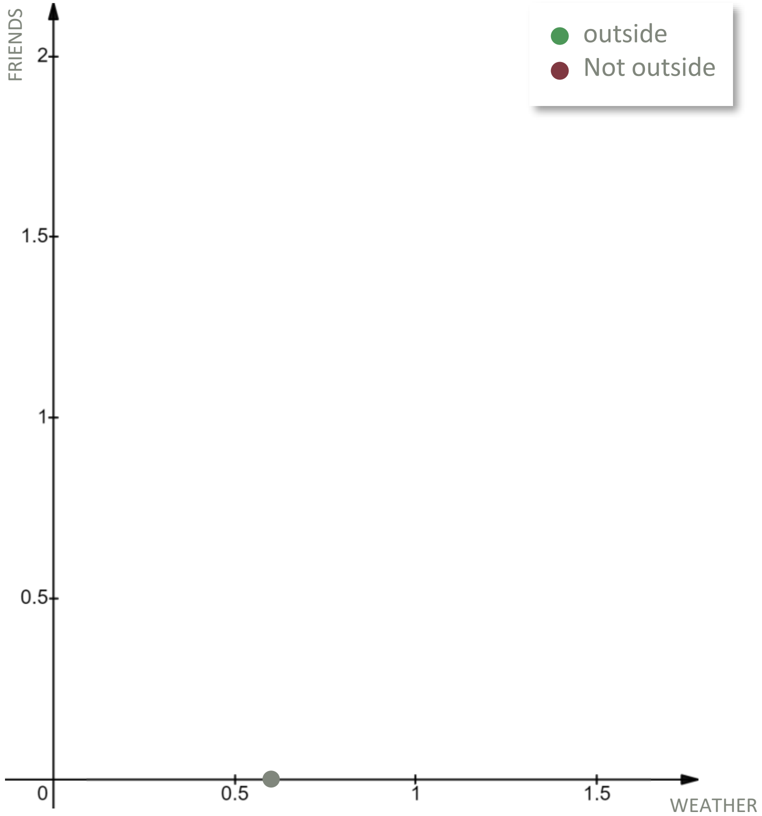
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



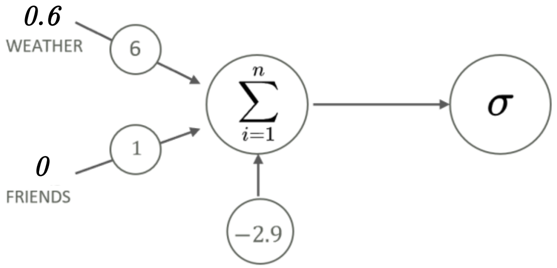
$$\sigma\left(-2.9+\sum_{i=1}^{n}x_i*\omega_i\right)$$

$$\sigma\left(\ 0.7\ \right)$$



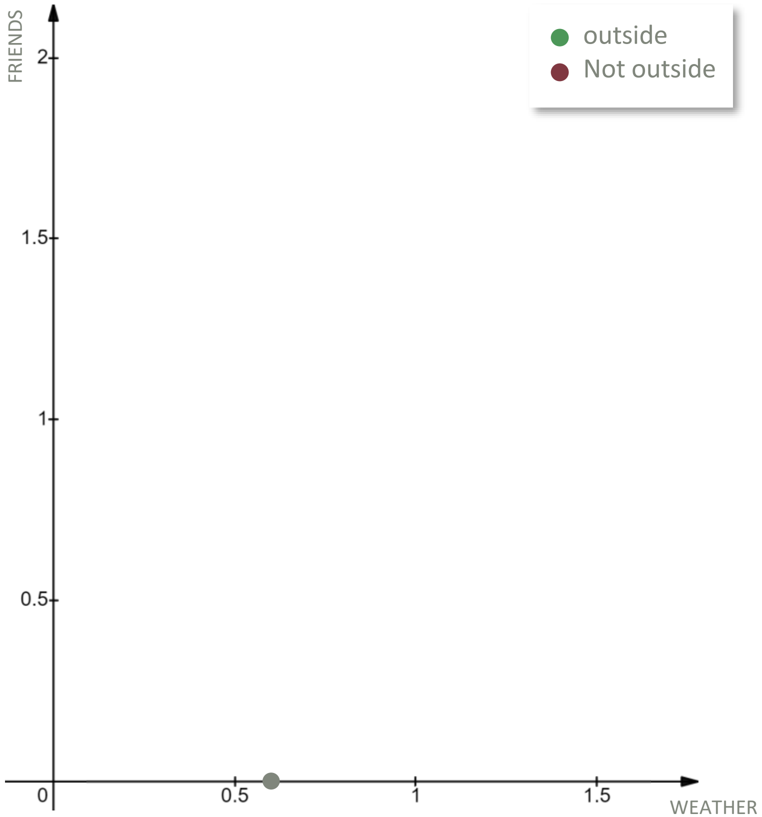
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



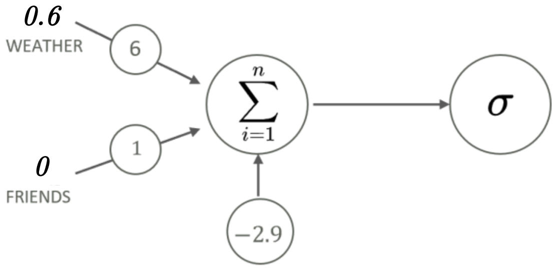
$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right)$$

$$\sigma\left(\ 0.7\ \right) = \frac{1}{(1+\ e^{-0.7})}$$



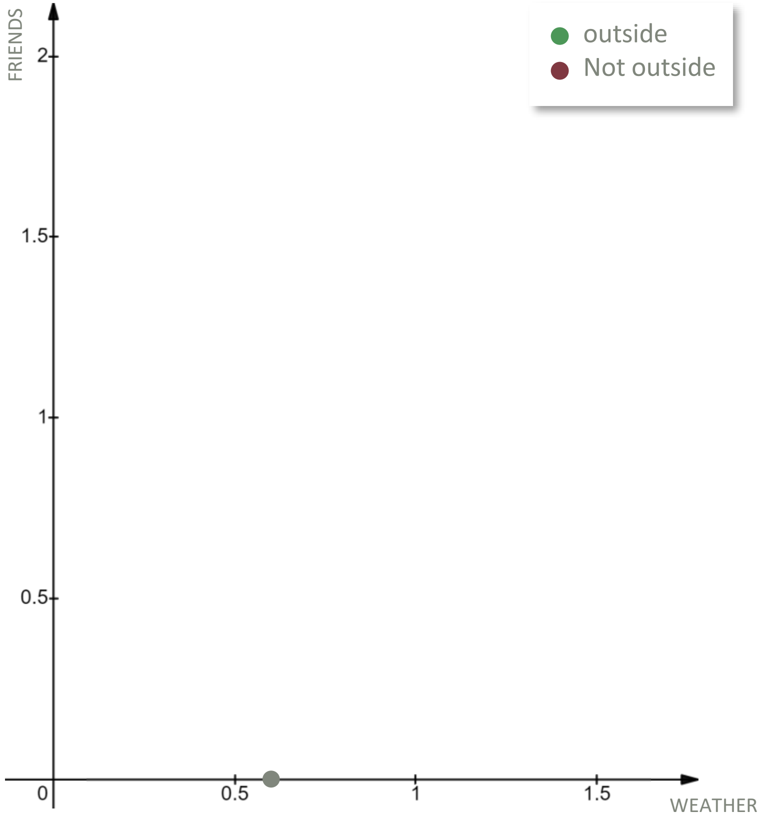
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



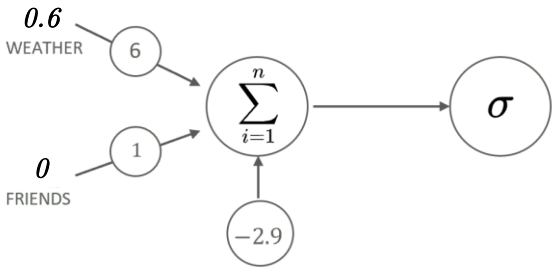
$$\sigma\left(-2.9 + \sum_{i=1}^{n} x_i * \omega_i\right)$$

$$\sigma\left(\ 0.7\ \right) = \frac{1}{(1+\ e^{-0.7})} = 0.668$$



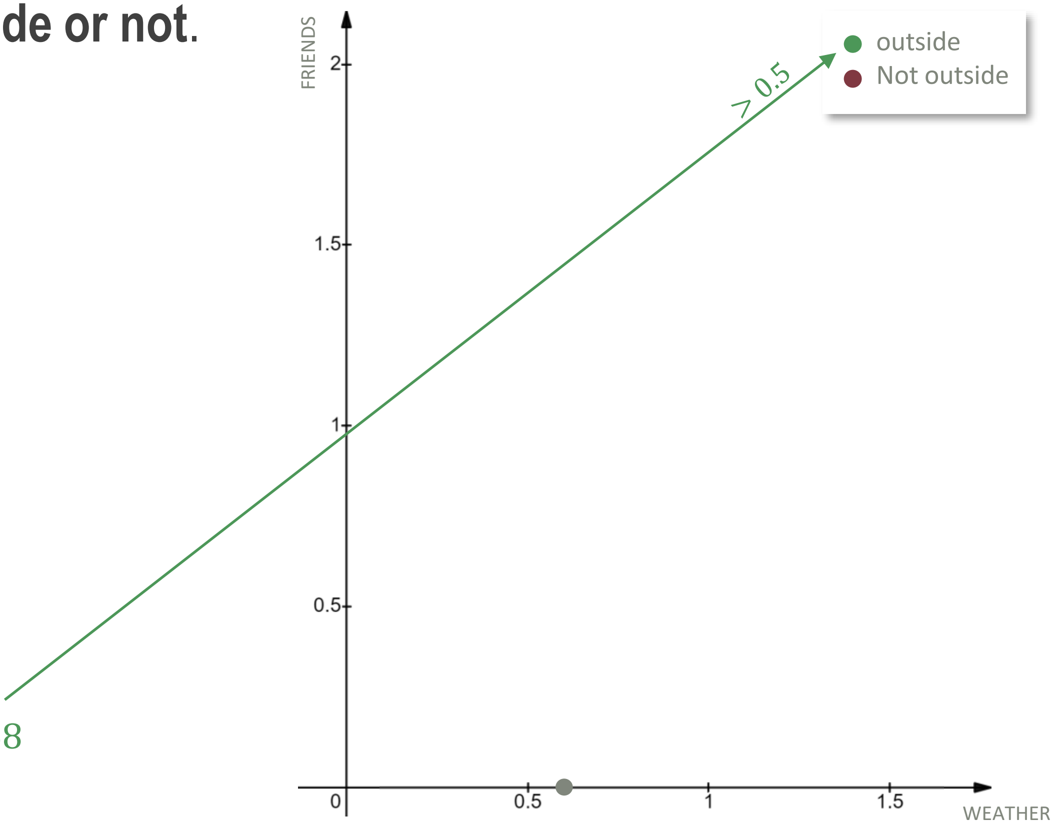
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



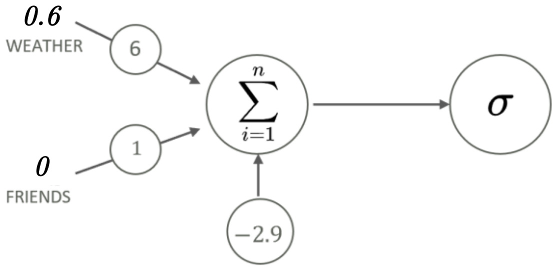
$$\sigma\left(-2.9 + \sum_{i=1}^{n} x_i * \omega_i\right)$$

$$\sigma\left(\ 0.7\ \right) = \frac{1}{(1 + \ e^{-0.7})} = 0.668$$



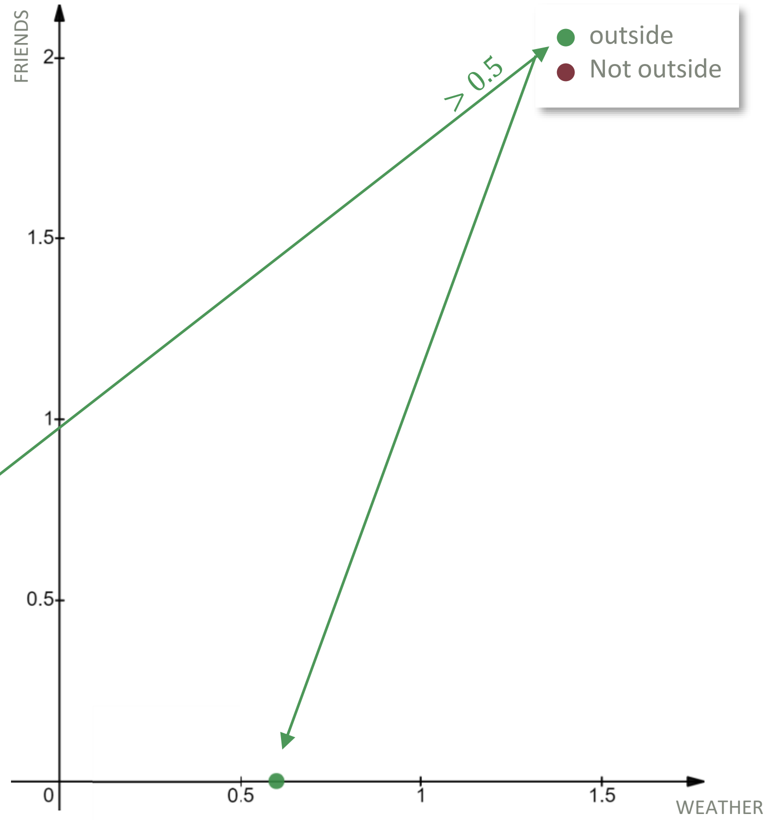
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



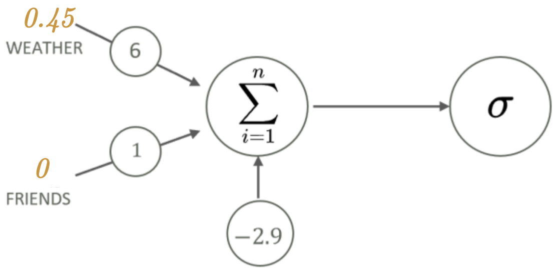
$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right)$$

$$\sigma\left(\ 0.7\ \right) = \frac{1}{(1+\ e^{-0.7})} = 0.668$$

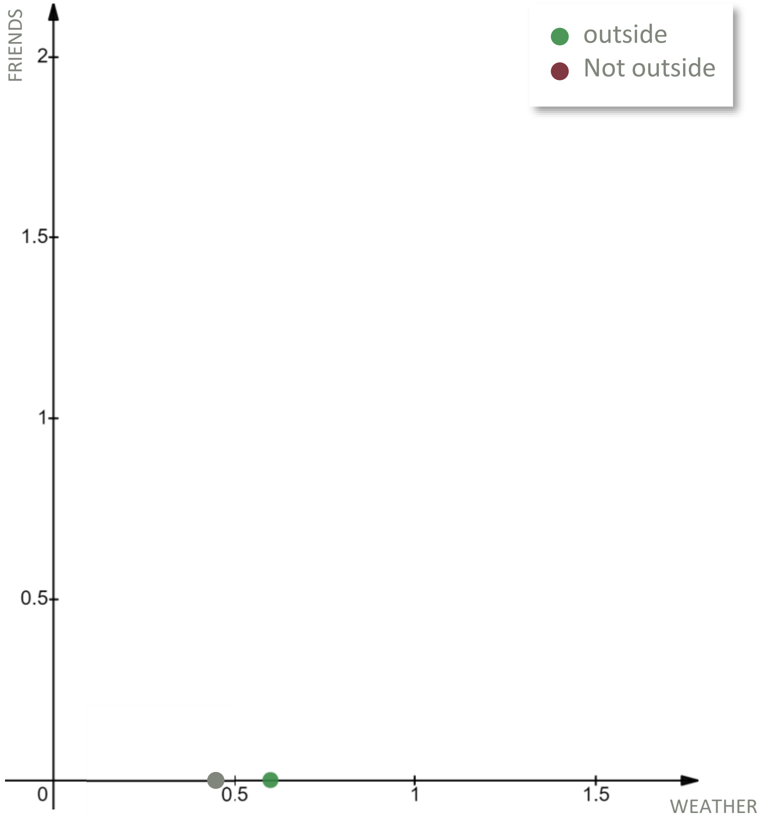


NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

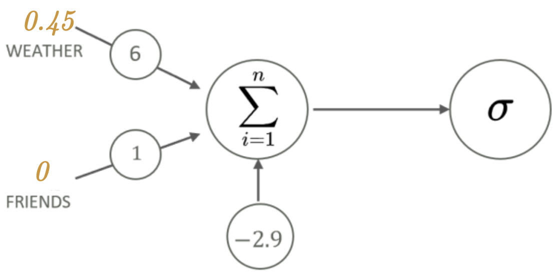


$$\sigma\left(-2.9+\sum_{i=1}^{n}x_i * \omega_i\right)$$



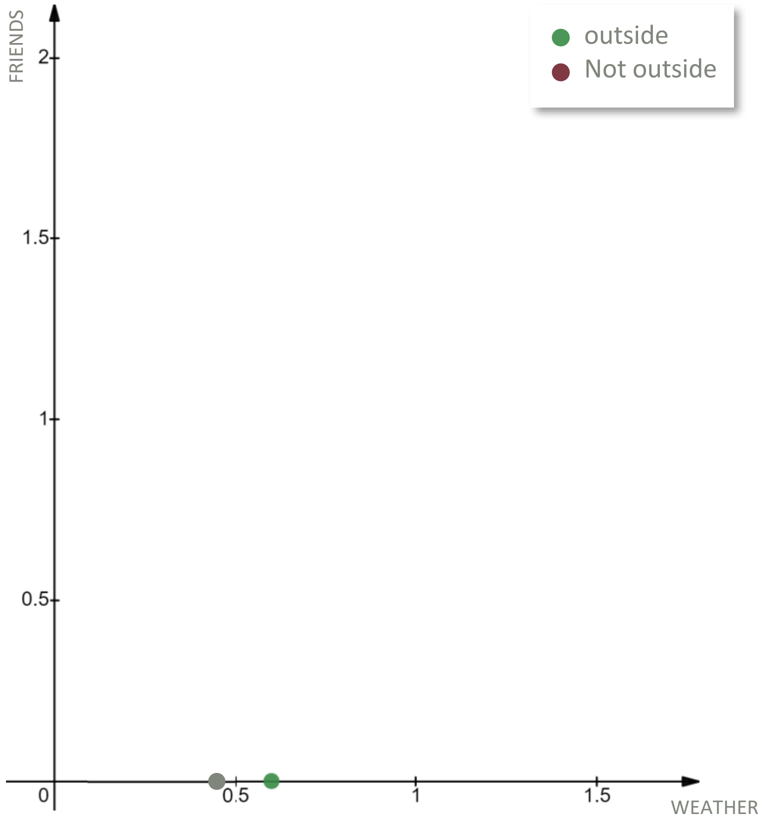
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



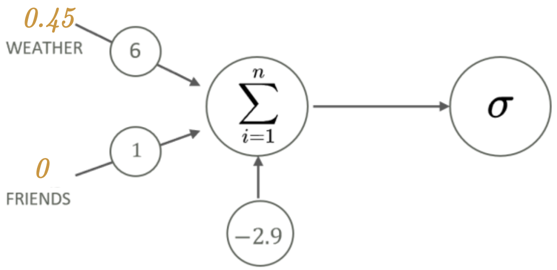
$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right)$$

$$\sigma\left(-2.9+0.45*6\ +\ 0\ \ *1\right)$$



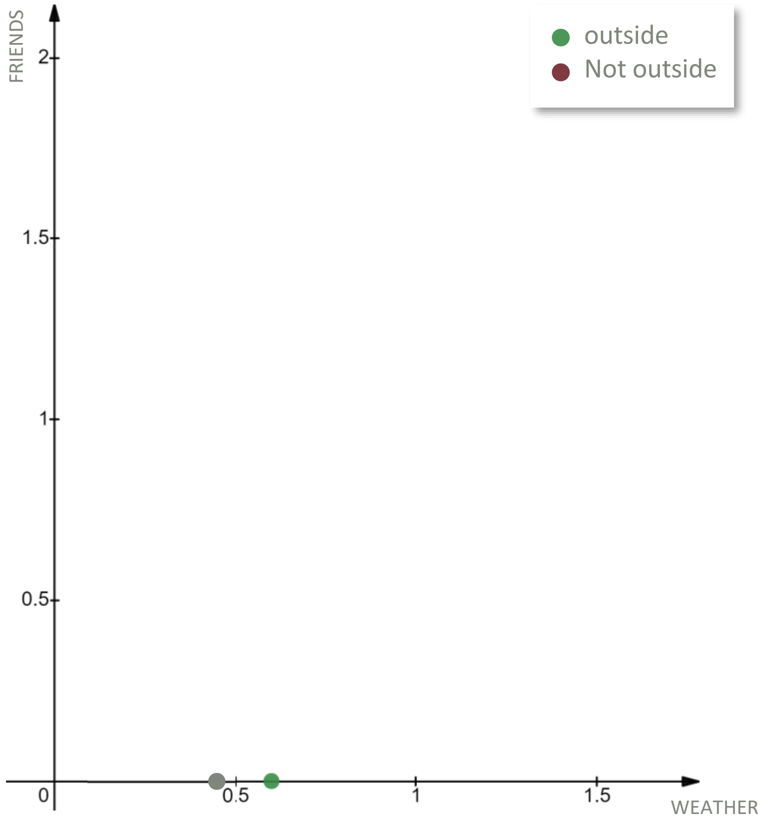
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



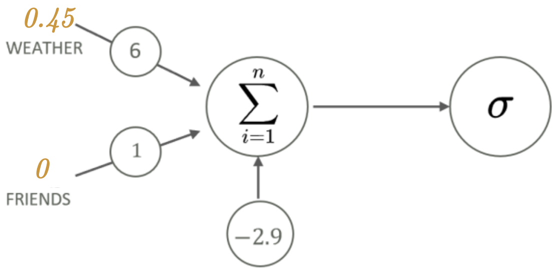
$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right)$$

$$\sigma\left(-2.9+0.45*6+0*1\right) = 0.450$$



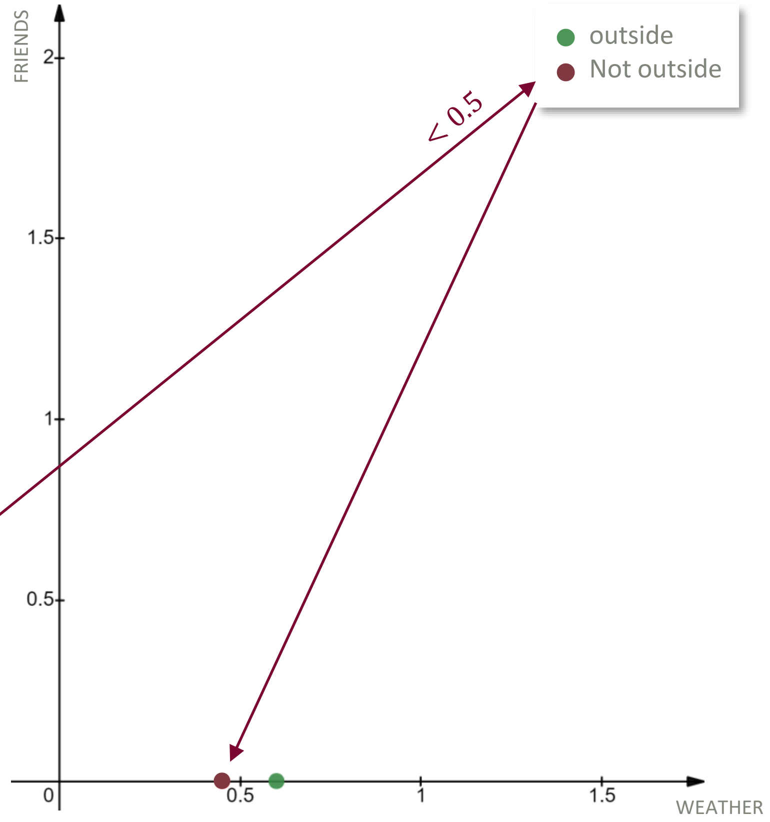
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

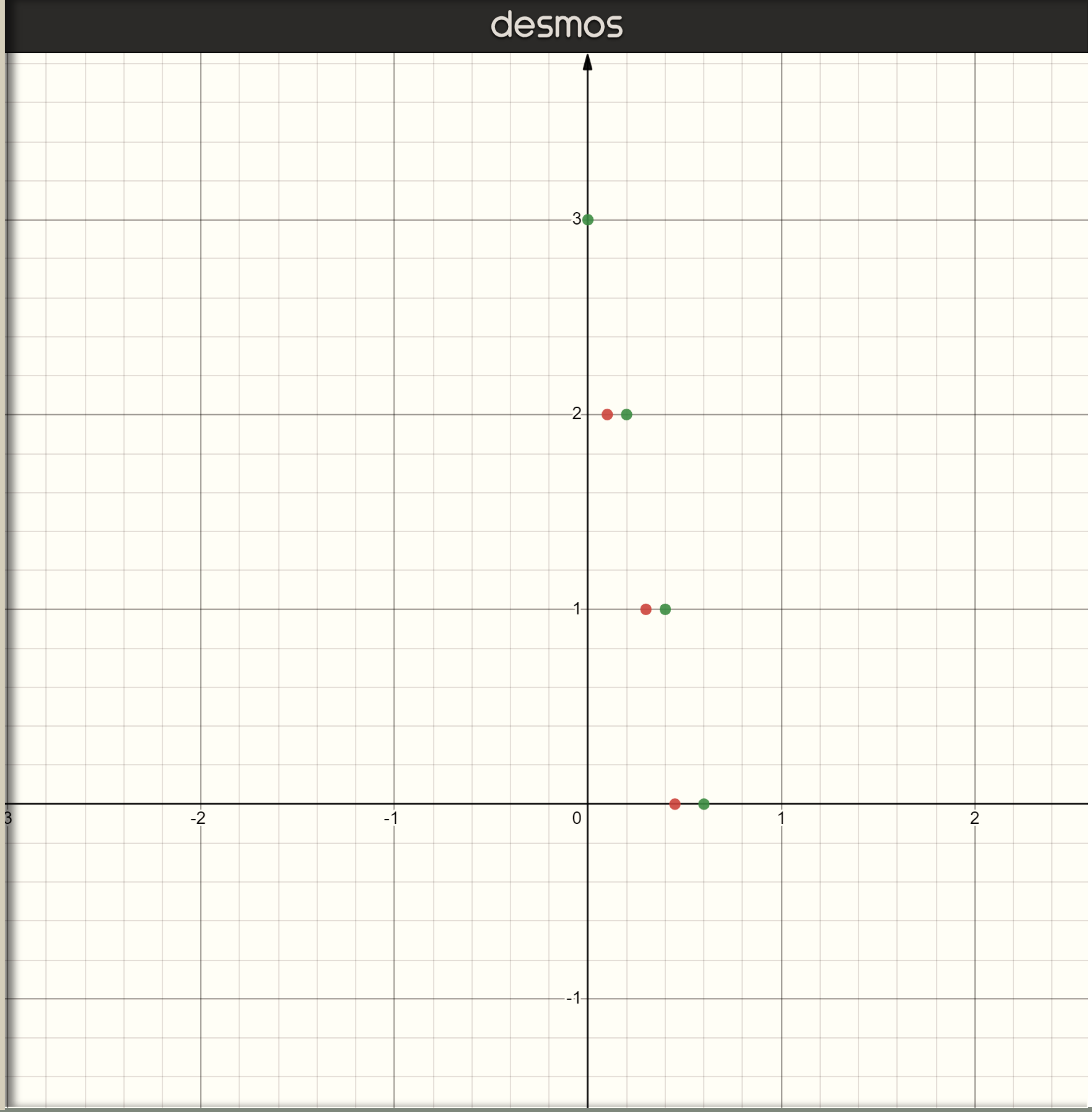


$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right)$$

$$\sigma\left(-2.9+0.45*6+0*1\right) = 0.450$$

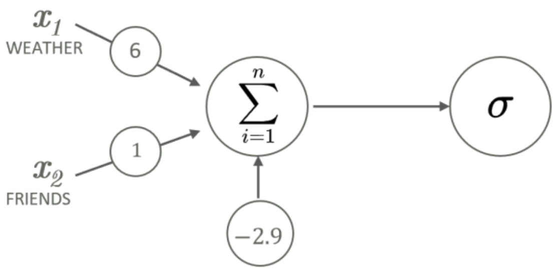


TIME FOR DESMOS

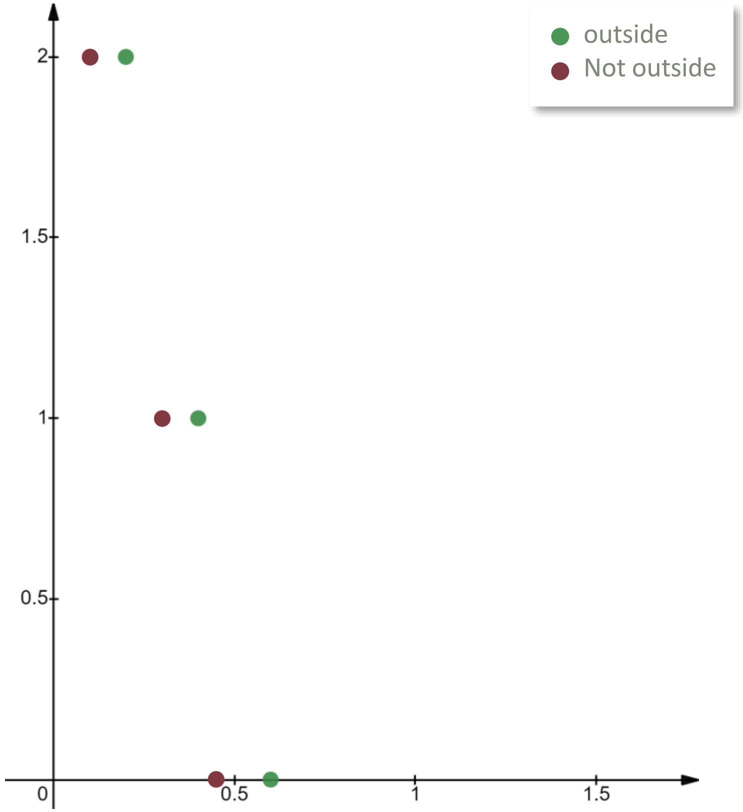


NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

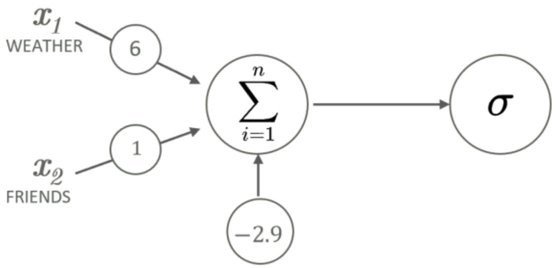


$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right)$$

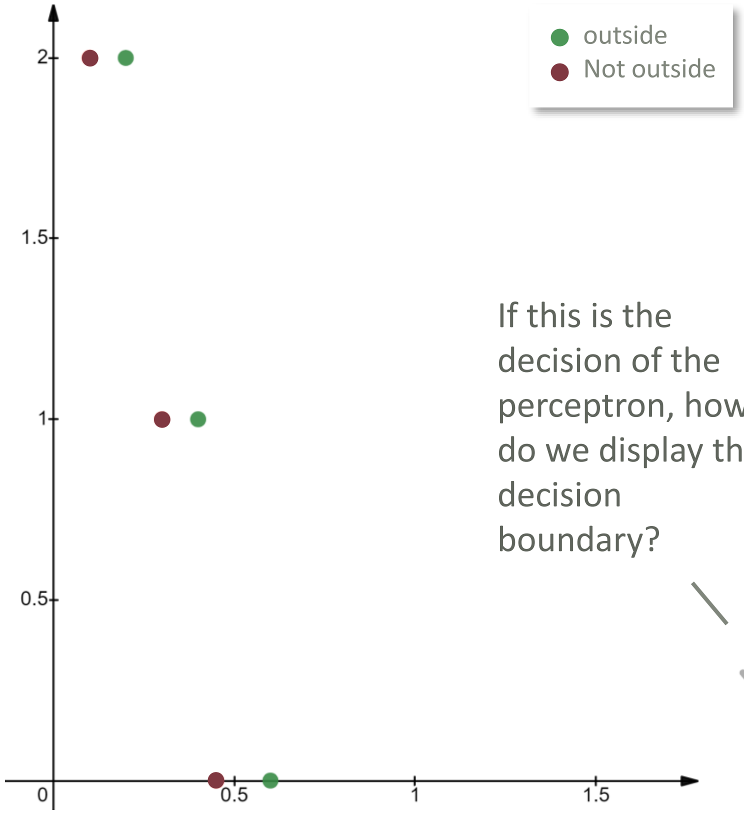


NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

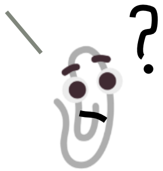


$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right)$$



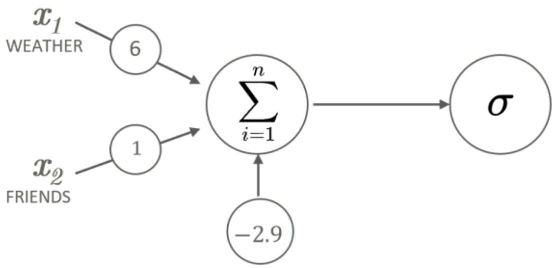
🟩 outside
🟥 Not outside

If this is the decision of the perceptron, how do we display the decision boundary?

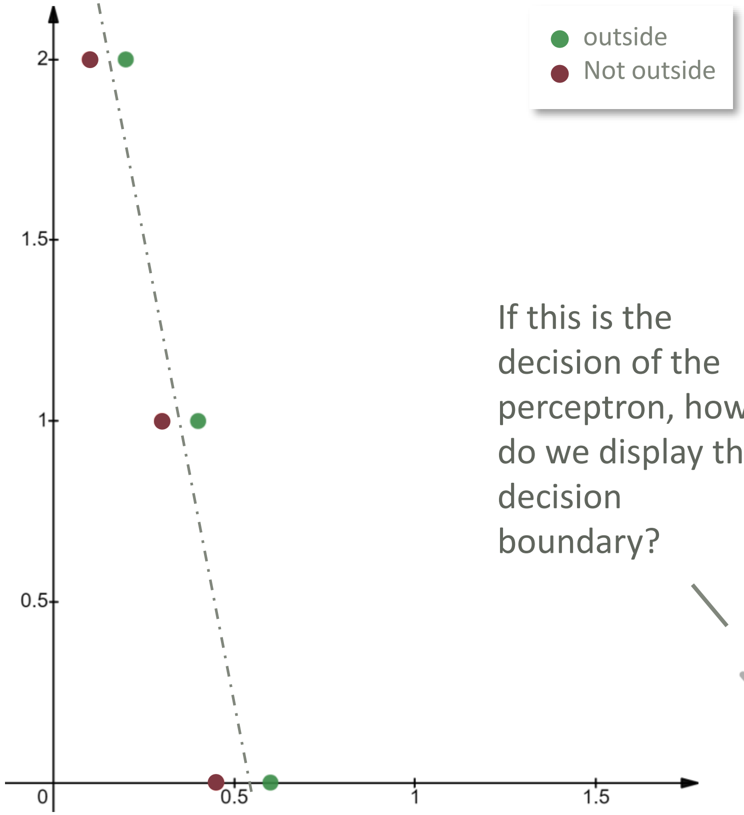


NEURAL NETWORK
A PERCEPTRON

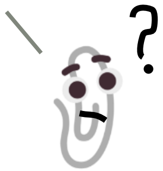
Ex: We want to decide to go **outside or not**.



$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right)$$

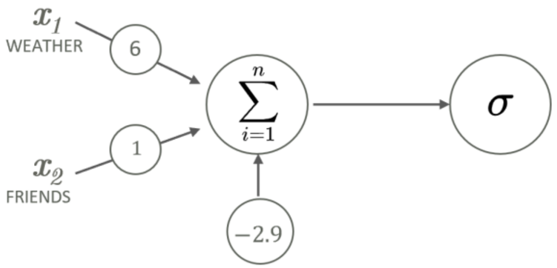


If this is the decision of the perceptron, how do we display the decision boundary?

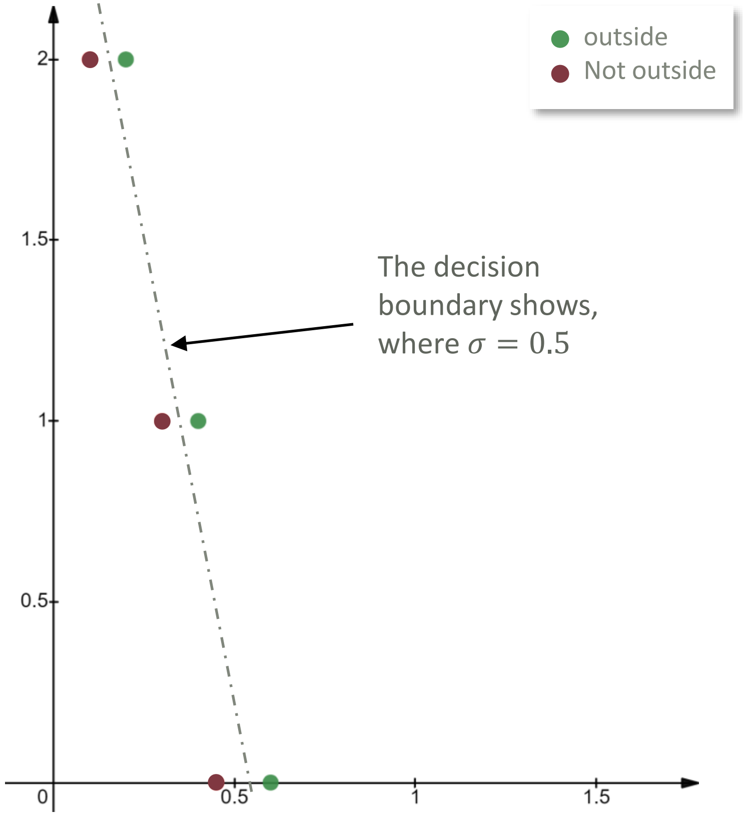


NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



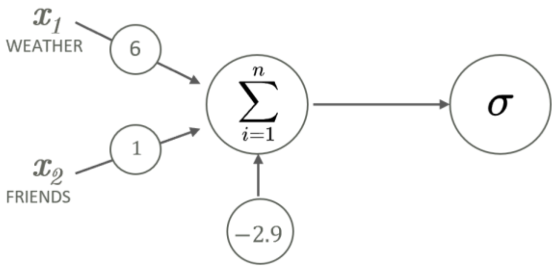
$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right)$$



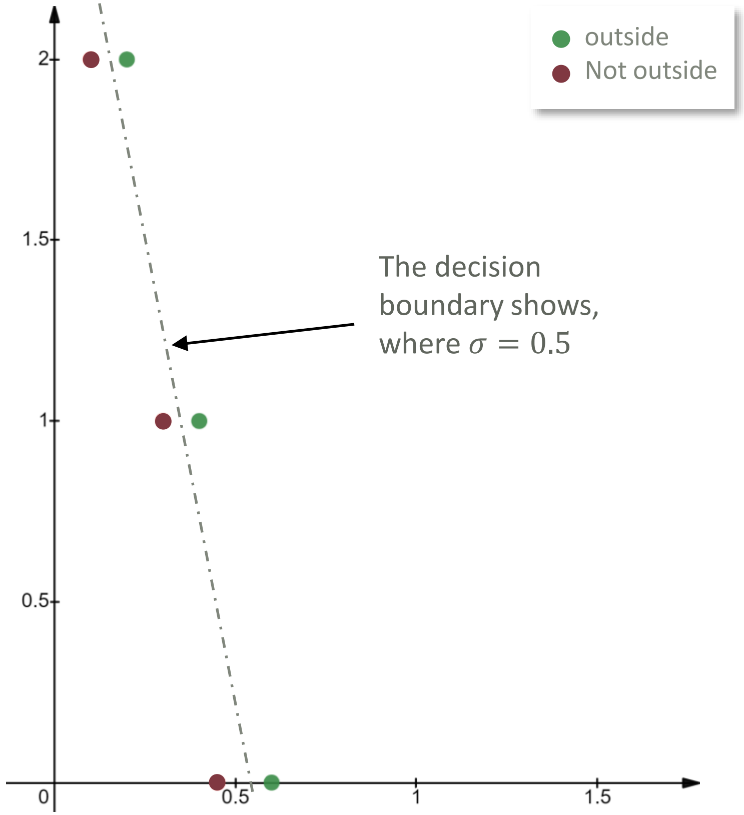
The decision boundary shows, where $\sigma = 0.5$

NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.



$$\sigma\left(-2.9 + \sum_{i=1}^{n} x_i * \omega_i\right) = 0.5$$

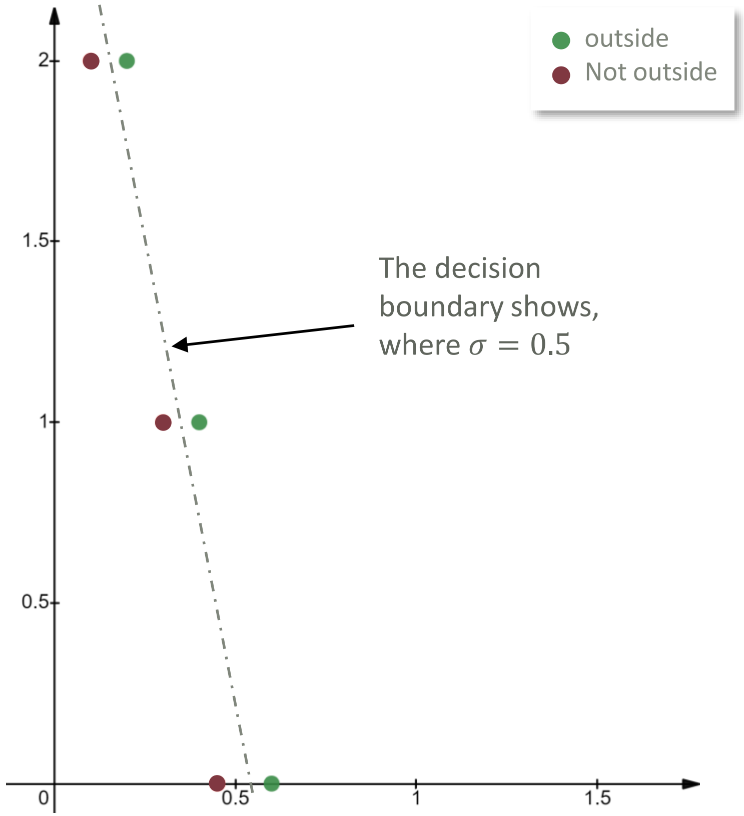


The decision boundary shows, where $\sigma = 0.5$

NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

$$\sigma\left(-2.9 + \sum_{i=1}^{n} x_i * \omega_i\right) = 0.5$$



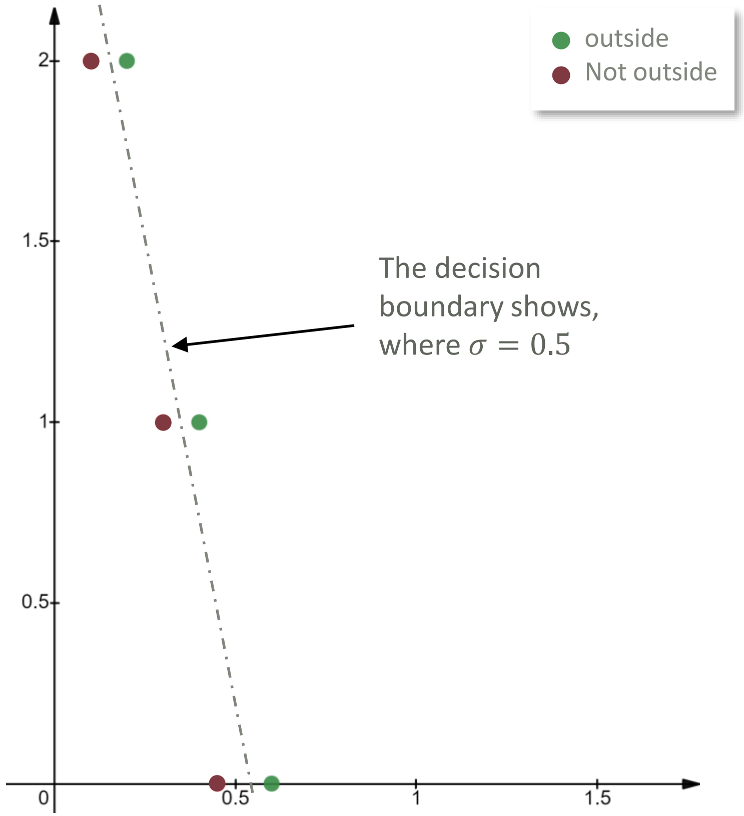
The decision boundary shows, where $\sigma = 0.5$

NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right)=0.5$$

$$\sigma\left(-2.9+(x_1 * 6+x_2 \ * 1)\right)=0.5$$



The decision boundary shows, where $\sigma = 0.5$

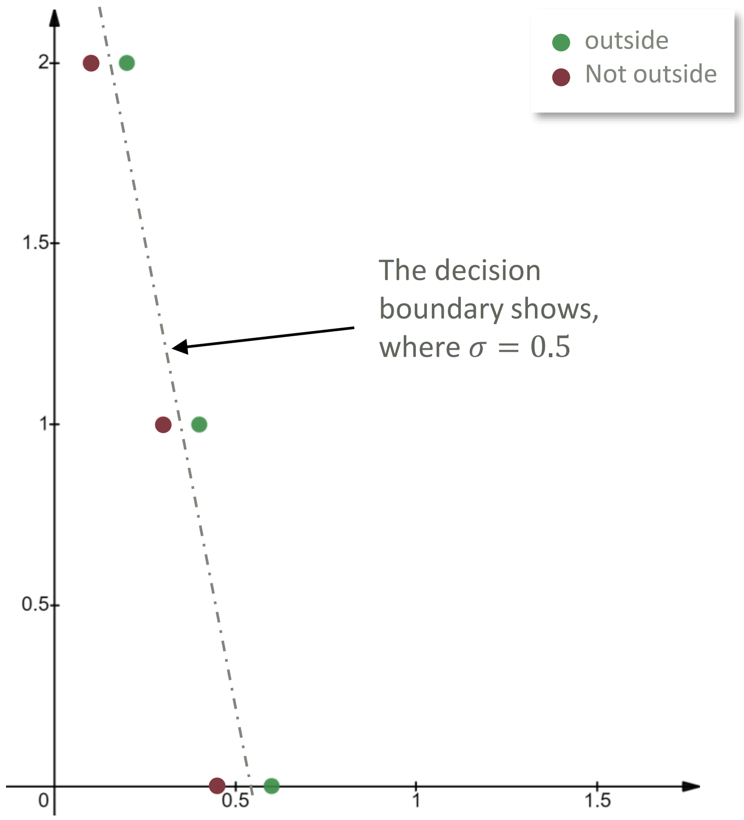
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

$$\sigma\left(-2.9 + \sum_{i=1}^{n} x_i * \omega_i\right) = 0.5$$

$$\sigma\left(-2.9 + (x_1 * 6 + x_2 \ * 1)\right) = 0.5$$

$$\frac{1}{1 + e^{-(-2.9+(x_1*6+x_2 \ *1))}} = 0.5$$



The decision boundary shows, where $\sigma = 0.5$

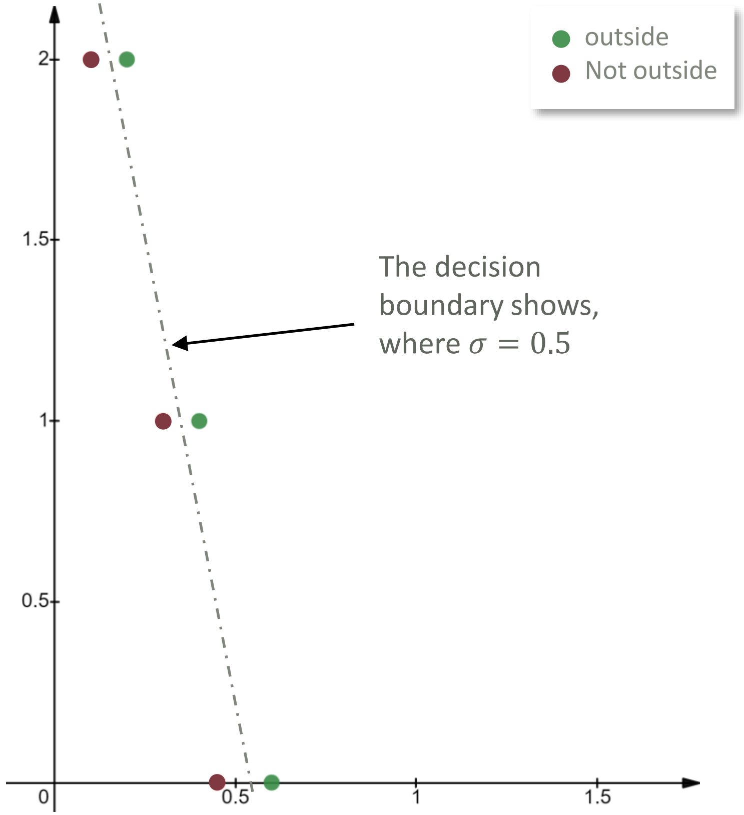
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

$$\sigma\left(-2.9 + \sum_{i=1}^{n} x_i * \omega_i\right) = 0.5$$

$$\sigma\left(-2.9 + (x_1 * 6 + x_2 \ * 1)\right) = 0.5$$

$$\frac{1}{1 + e^{-(-2.9+(6x_1+x_2))}} = 0.5$$



The decision boundary shows, where $\sigma = 0.5$

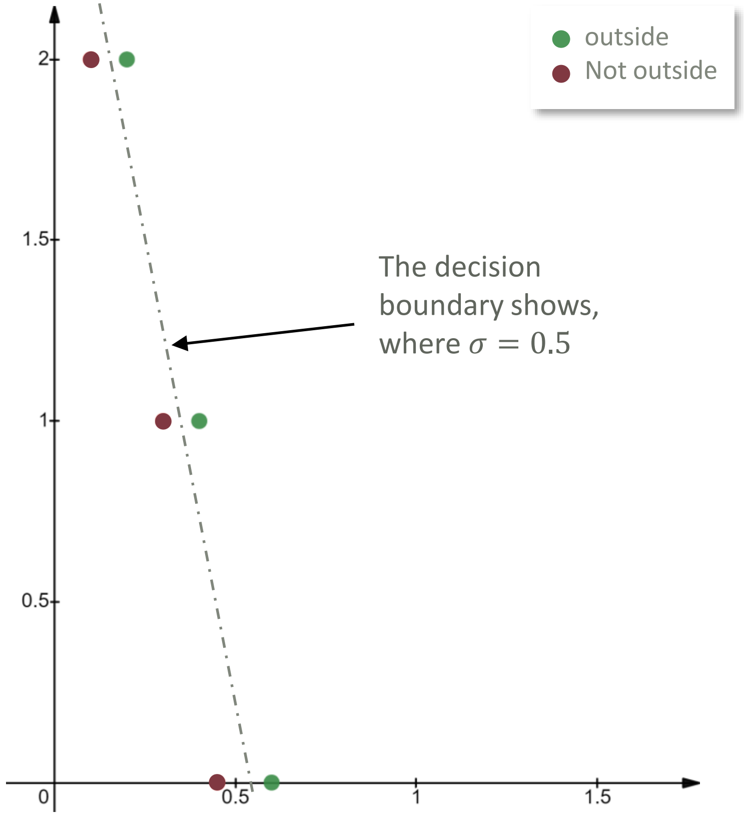
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

$$\sigma\left(-2.9 + \sum_{i=1}^{n} x_i * \omega_i\right) = 0.5$$

$$\sigma\left(-2.9 + (x_1 * 6 + x_2 \ * 1)\right) = 0.5$$

$$\frac{1}{1 + e^{(2.9-(6x_1+x_2))}} = 0.5$$



The decision boundary shows, where $\sigma = 0.5$

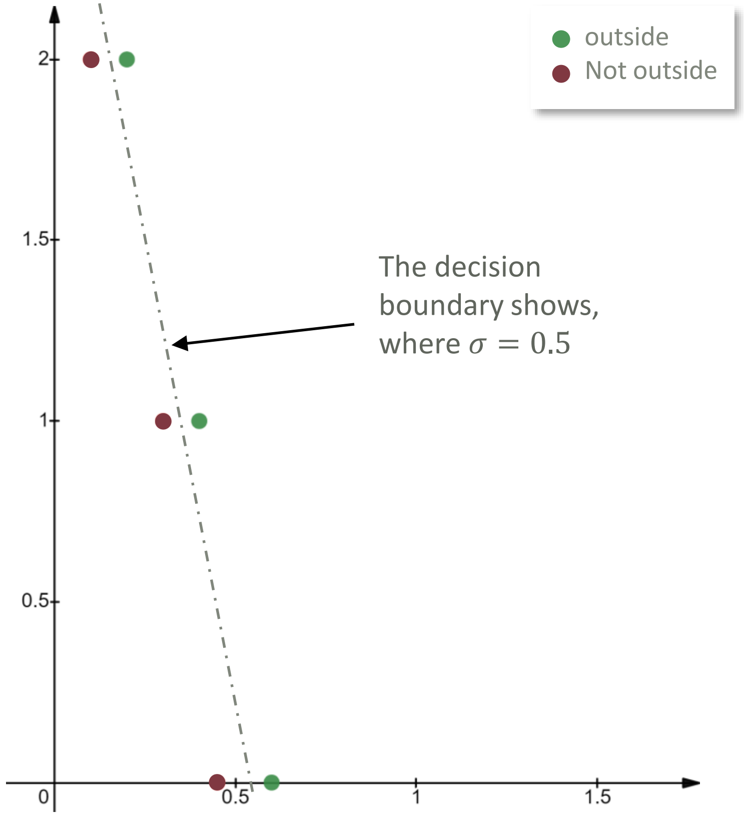
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

$$\sigma\left(-2.9 + \sum_{i=1}^{n} x_i * \omega_i\right) = 0.5$$

$$\sigma\left(-2.9 + (x_1 * 6 + x_2 \ * 1)\right) = 0.5$$

$$\frac{1}{1 + e^{(2.9-(6x_1+x_2))}} = \frac{1}{2}$$



The decision boundary shows, where $\sigma = 0.5$

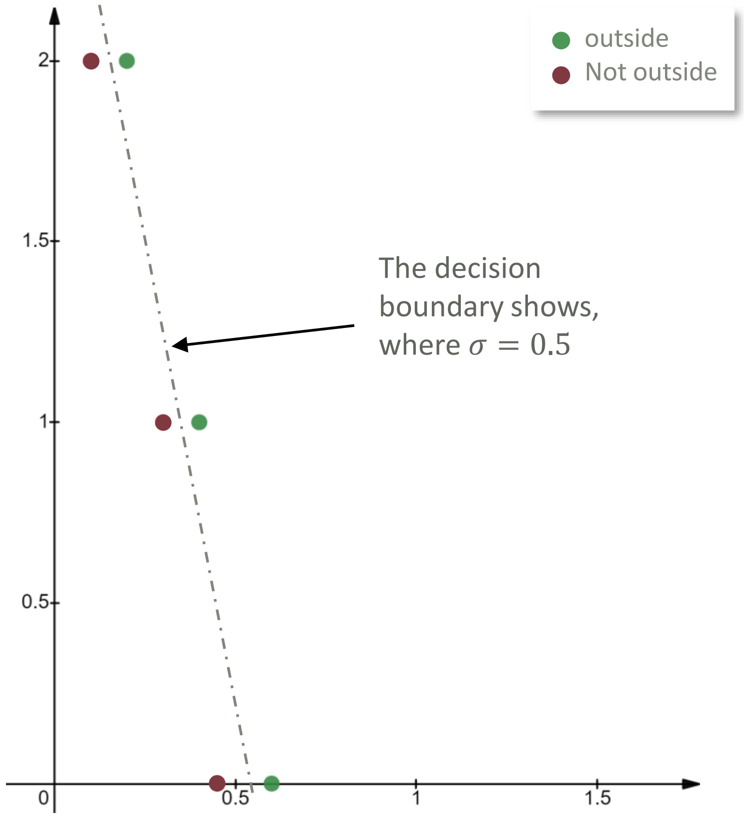
NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right) = 0.5$$

$$\sigma\left(-2.9+(x_1 * 6 + x_2 \ * 1)\right) = 0.5$$

$$\frac{\cancel{1}}{1+e^{(2.9-(6x_1+x_2))}} = \frac{\cancel{1}}{2}$$



The decision boundary shows, where $\sigma = 0.5$

NEURAL NETWORK
A PERCEPTRON

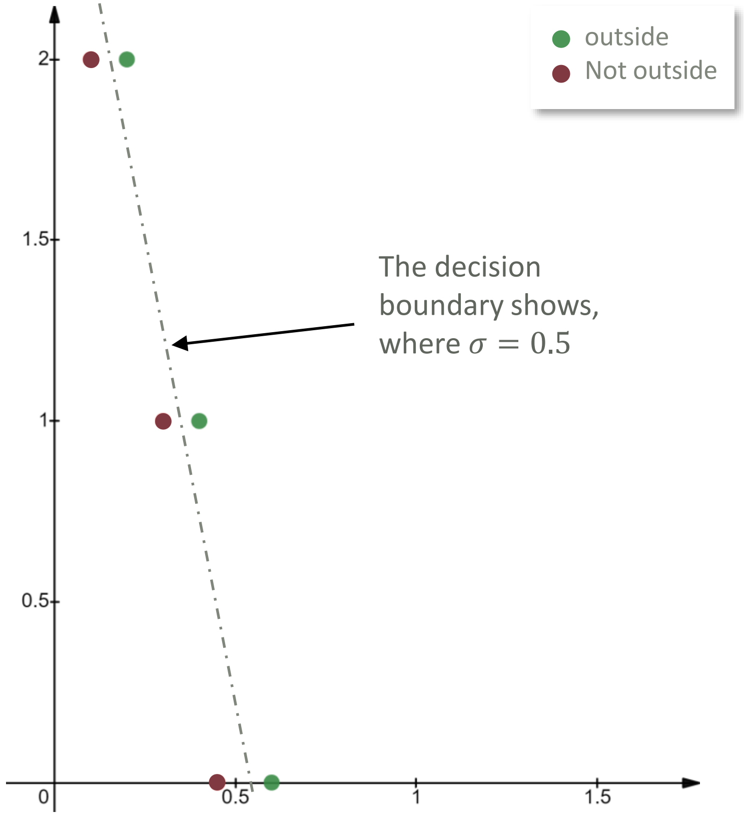
Ex: We want to decide to go **outside or not**.

$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right) = 0.5$$

$$\sigma\left(-2.9+(x_1 * 6 + x_2 \ * 1)\right) = 0.5$$

$$\frac{\cancel{1}}{1+e^{(2.9-(6x_1+x_2))}} = \frac{\cancel{1}}{2}$$

$$1+e^{(2.9-(6x_1+x_2))} = 2$$



The decision boundary shows, where $\sigma = 0.5$

NEURAL NETWORK
A PERCEPTRON

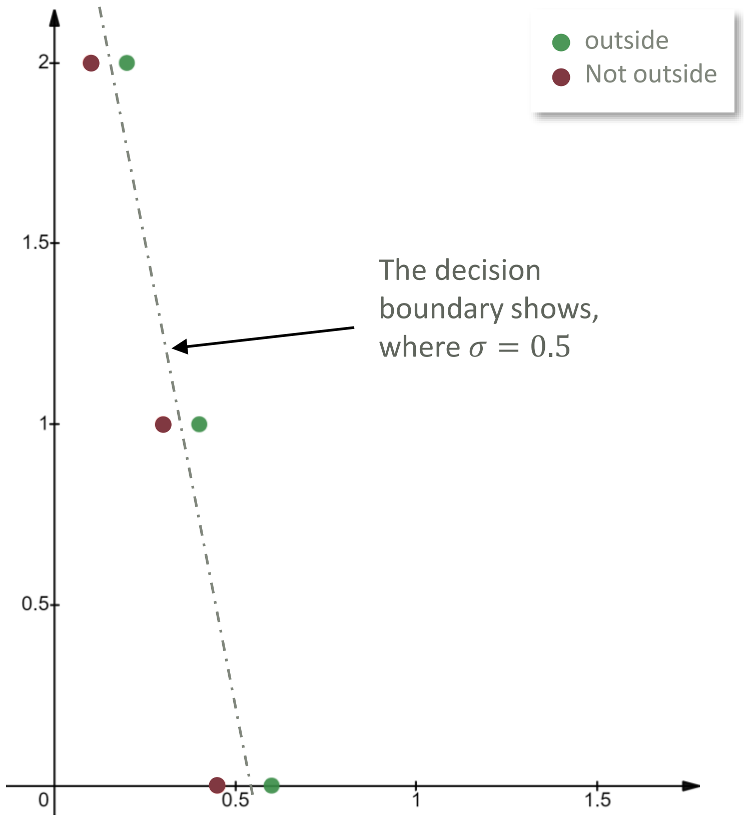
Ex: We want to decide to go **outside or not**.

$$\sigma\left(-2.9 + \sum_{i=1}^{n} x_i * \omega_i\right) = 0.5$$

$$\sigma\left(-2.9 + (x_1 * 6 + x_2 \ * 1)\right) = 0.5$$

$$\frac{\cancel{1}}{1 + e^{(2.9-(6x_1+x_2))}} = \frac{\cancel{1}}{2}$$

$$1 + e^{(2.9-6x_1-x_2)} = 2$$



The decision boundary shows, where $\sigma = 0.5$

NEURAL NETWORK
A PERCEPTRON

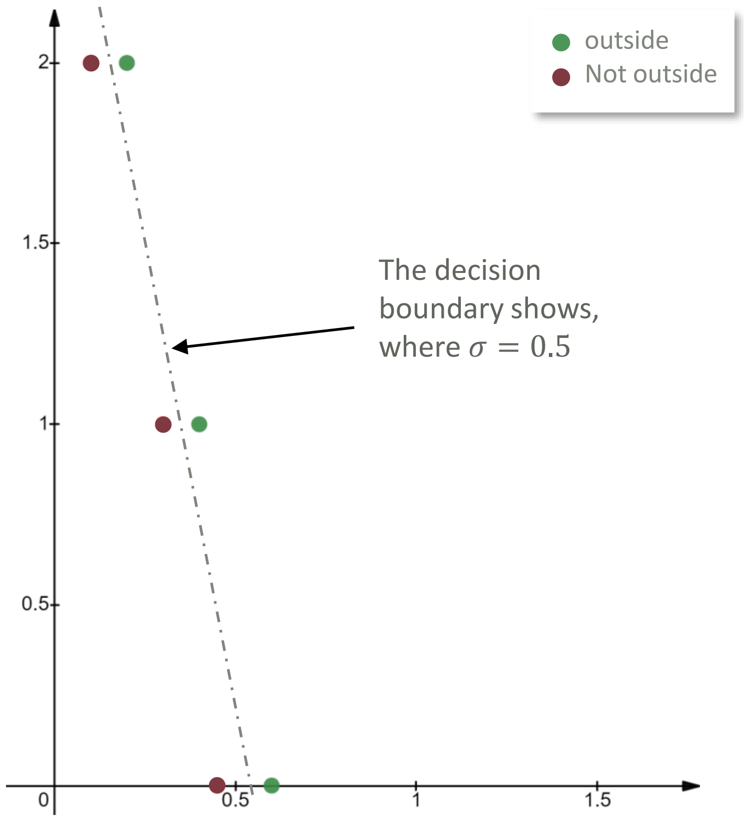
Ex: We want to decide to go **outside or not**.

$$\sigma\left(-2.9 + \sum_{i=1}^{n} x_i * \omega_i\right) = 0.5$$

$$\sigma\left(-2.9 + (x_1 * 6 + x_2 \ * 1)\right) = 0.5$$

$$\frac{\cancel{1}}{1 + e^{(2.9-(6x_1+x_2))}} = \frac{\cancel{1}}{2}$$

$$1 + e^{(2.9-6x_1-x_2)} = 2 \quad \Big| -1$$



The decision boundary shows, where $\sigma = 0.5$

NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

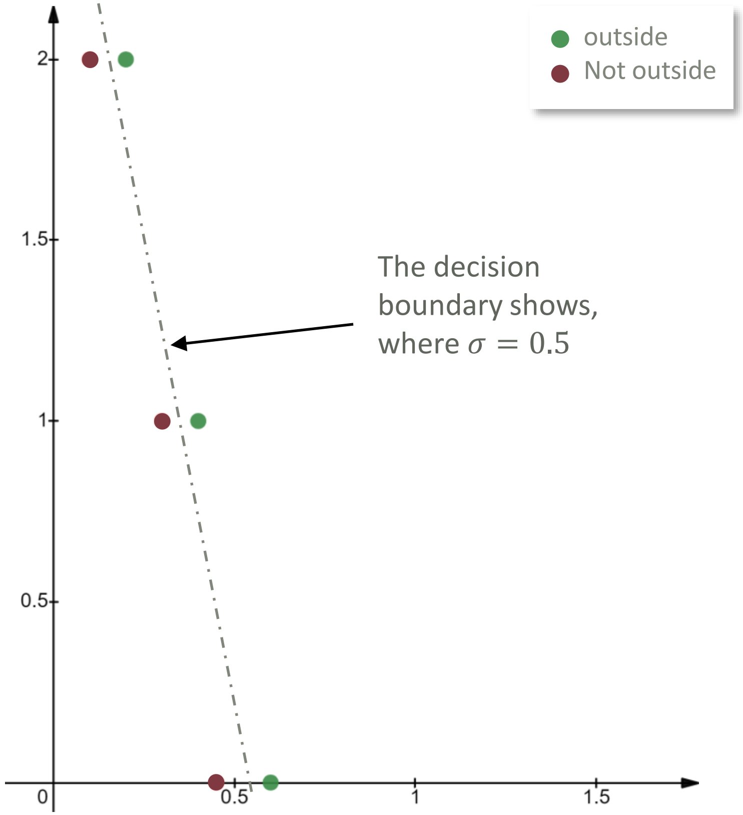
$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right) = 0.5$$

$$\sigma\left(-2.9+(x_1 * 6 + x_2 \ * 1)\right) = 0.5$$

$$\frac{\cancel{1}}{1+e^{(2.9-(6x_1+x_2))}} = \frac{\cancel{1}}{2}$$

$$1+e^{(2.9-6x_1-x_2)} = 2 \quad \Big| \ -1$$

$$e^{(2.9-6x_1-x_2)} = 1$$



The decision boundary shows, where $\sigma = 0.5$

NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

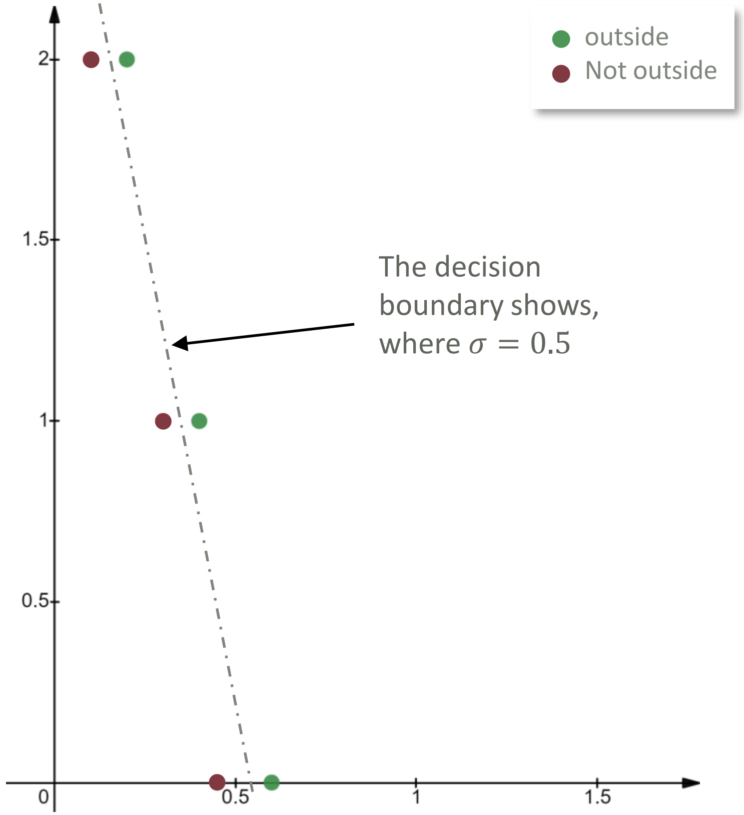
$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right) = 0.5$$

$$\sigma\left(-2.9+(x_1 * 6 + x_2 \ * 1)\right) = 0.5$$

$$\frac{\cancel{1}}{1+e^{(2.9-(6x_1+x_2))}} = \frac{\cancel{1}}{2}$$

$$1+e^{(2.9-6x_1-x_2)} = 2 \quad \Big| \ -1$$

$$e^{(2.9-6x_1-x_2)} = 1 \quad \Big| \ \log()$$



The decision boundary shows, where $\sigma = 0.5$

NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right) = 0.5$$

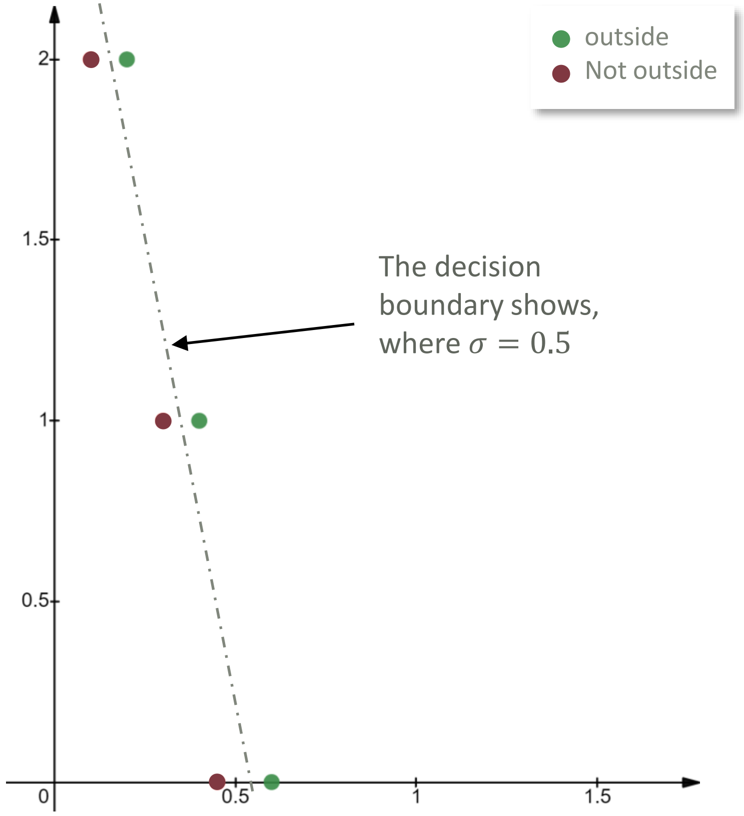
$$\sigma\left(-2.9+(x_1 * 6 + x_2 \ * 1)\right) = 0.5$$

$$\frac{\cancel{1}}{1+e^{(2.9-(6x_1+x_2))}} = \frac{\cancel{1}}{2}$$

$$1+e^{(2.9-6x_1-x_2)} = 2 \quad \Big| \ -1$$

$$e^{(2.9-6x_1-x_2)} = 1 \quad \Big| \ \log()$$

$$2.9-6x_1-x_2 = 0$$



The decision boundary shows, where $\sigma = 0.5$

NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right) = 0.5$$

$$\sigma\left(-2.9+(x_1 * 6 + x_2 \ * 1)\right) = 0.5$$

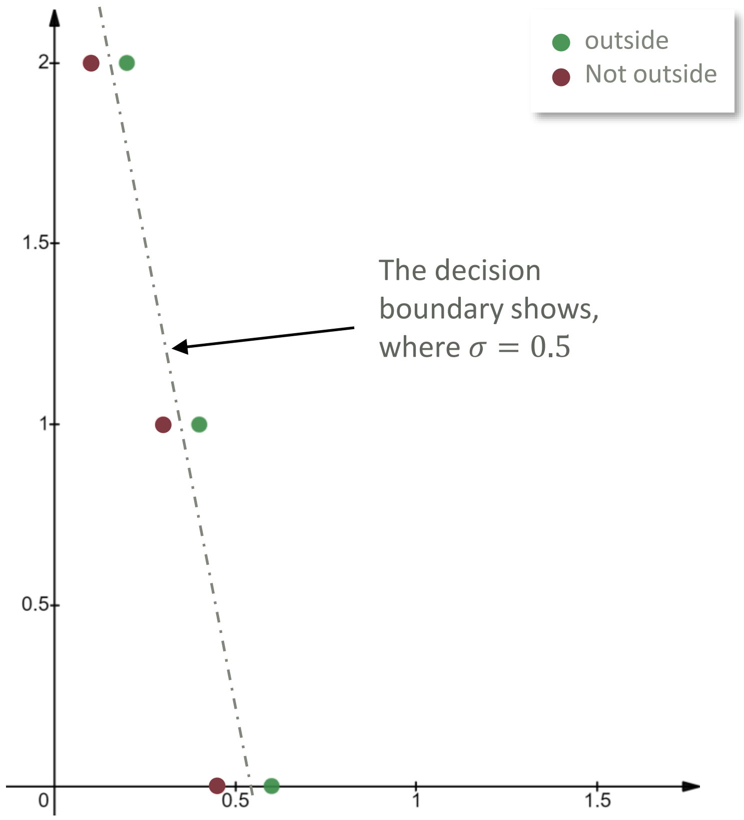
$$\frac{\cancel{1}}{1+e^{(2.9-(6x_1+x_2))}} = \frac{\cancel{1}}{2}$$

$$1+e^{(2.9-6x_1-x_2)} = 2 \quad \Big| \ -1$$

$$e^{(2.9-6x_1-x_2)} = 1 \quad \Big| \ \log()$$

$$2.9-6x_1-x_2 = 0$$

$$\underline{\underline{x_2 = 2.9-6x_1}}$$



The decision boundary shows, where $\sigma = 0.5$

NEURAL NETWORK
A PERCEPTRON

Ex: We want to decide to go **outside or not**.

$$\sigma\left(-2.9+\sum_{i=1}^{n} x_i * \omega_i\right) = 0.5$$

$$\sigma\left(-2.9+(x_1 * 6+x_2 \; * 1)\right) = 0.5$$

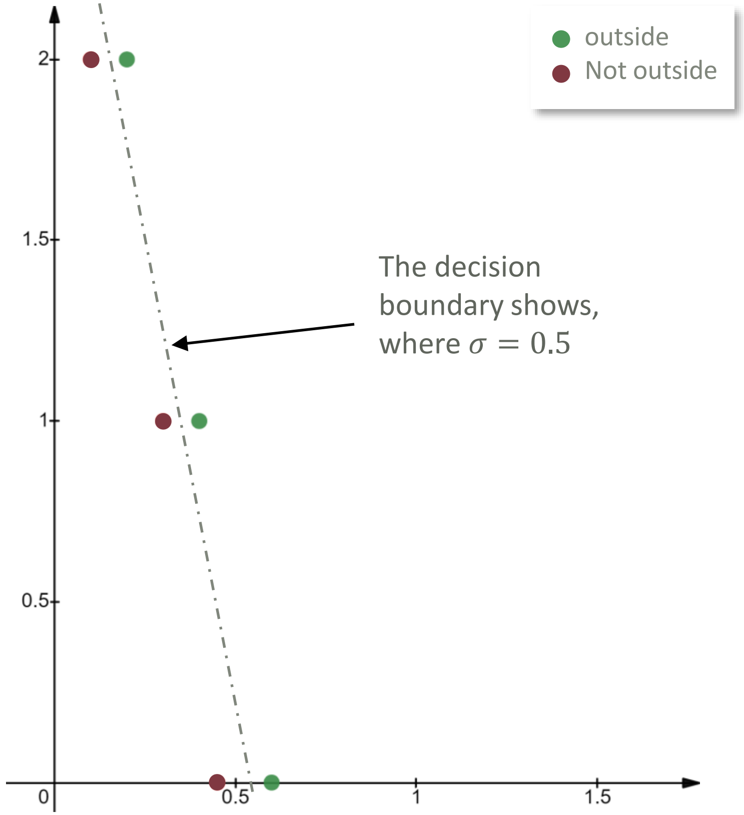
$$\frac{\cancel{1}}{1+e^{(2.9-(6x_1+x_2))}} = \frac{\cancel{1}}{2}$$

$$1+e^{(2.9-6x_1-x_2)} = 2 \quad \Big| \; -1$$

$$e^{(2.9-6x_1-x_2)} = 1 \quad \Big| \; \log()$$

$$2.9-6x_1-x_2 = 0$$

$$\underline{\underline{y = 2.9-6x}}$$

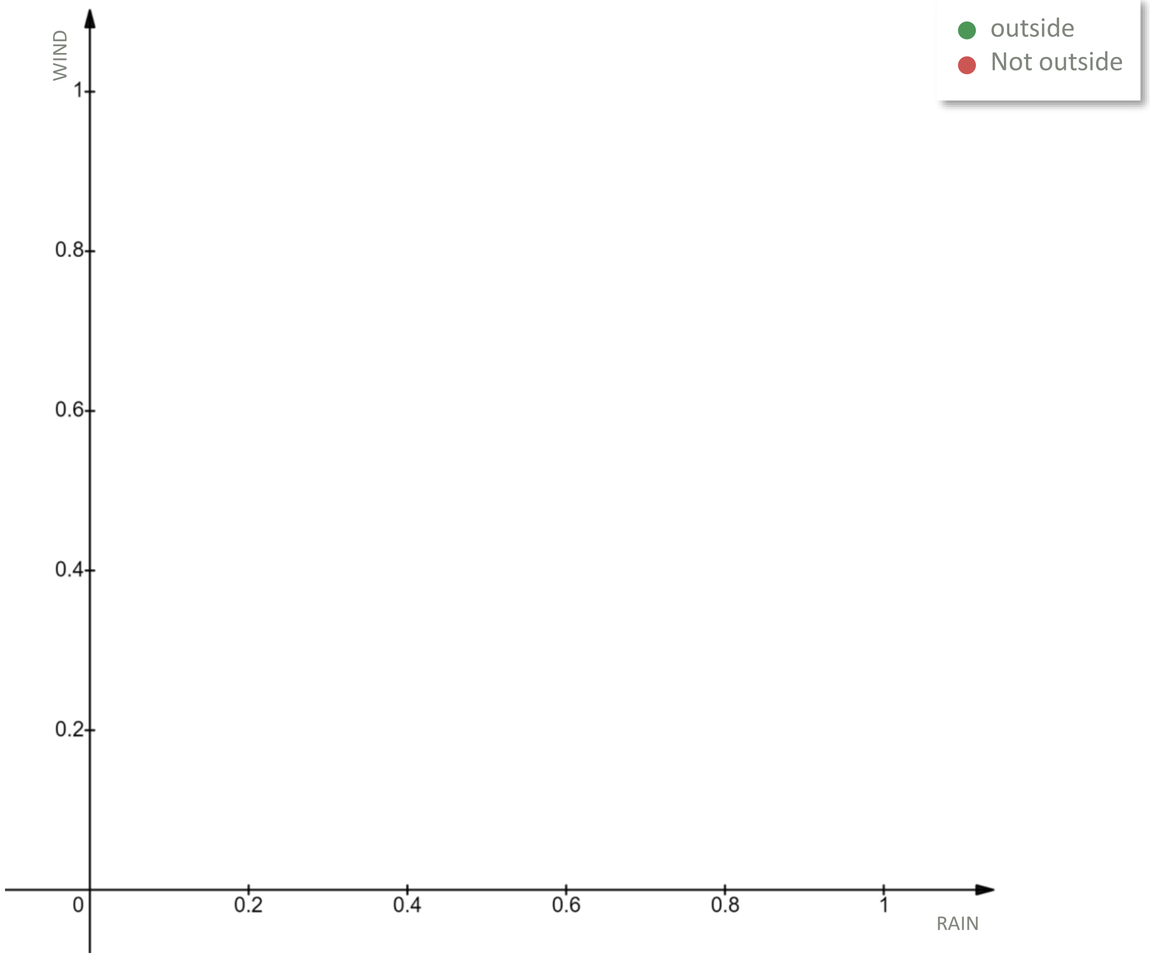


The decision boundary shows, where $\sigma = 0.5$

BUT WHAT ABOUT **NON-LINEAR** DECISIONS?

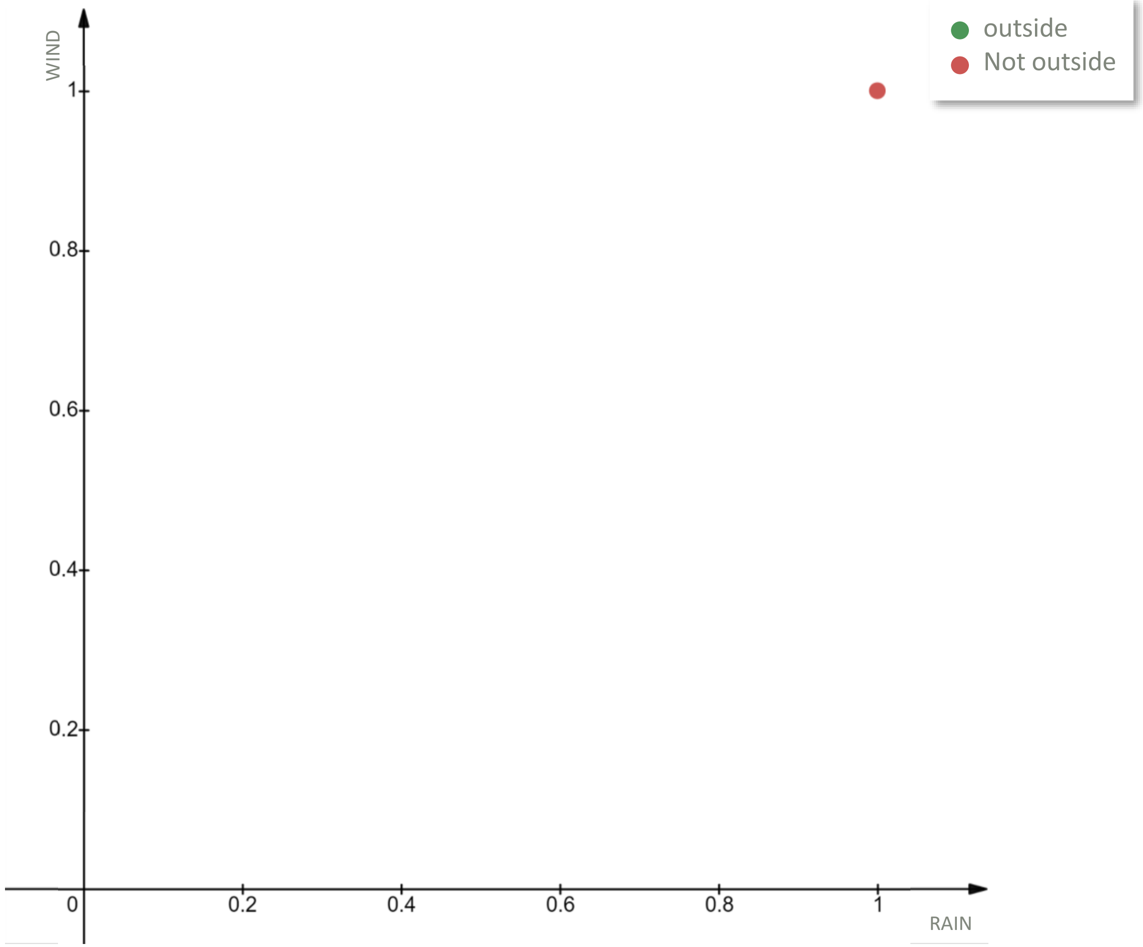
NEURAL NETWORK
MULTI-LAYER PERCEPTRON

Rain	Wind	Label



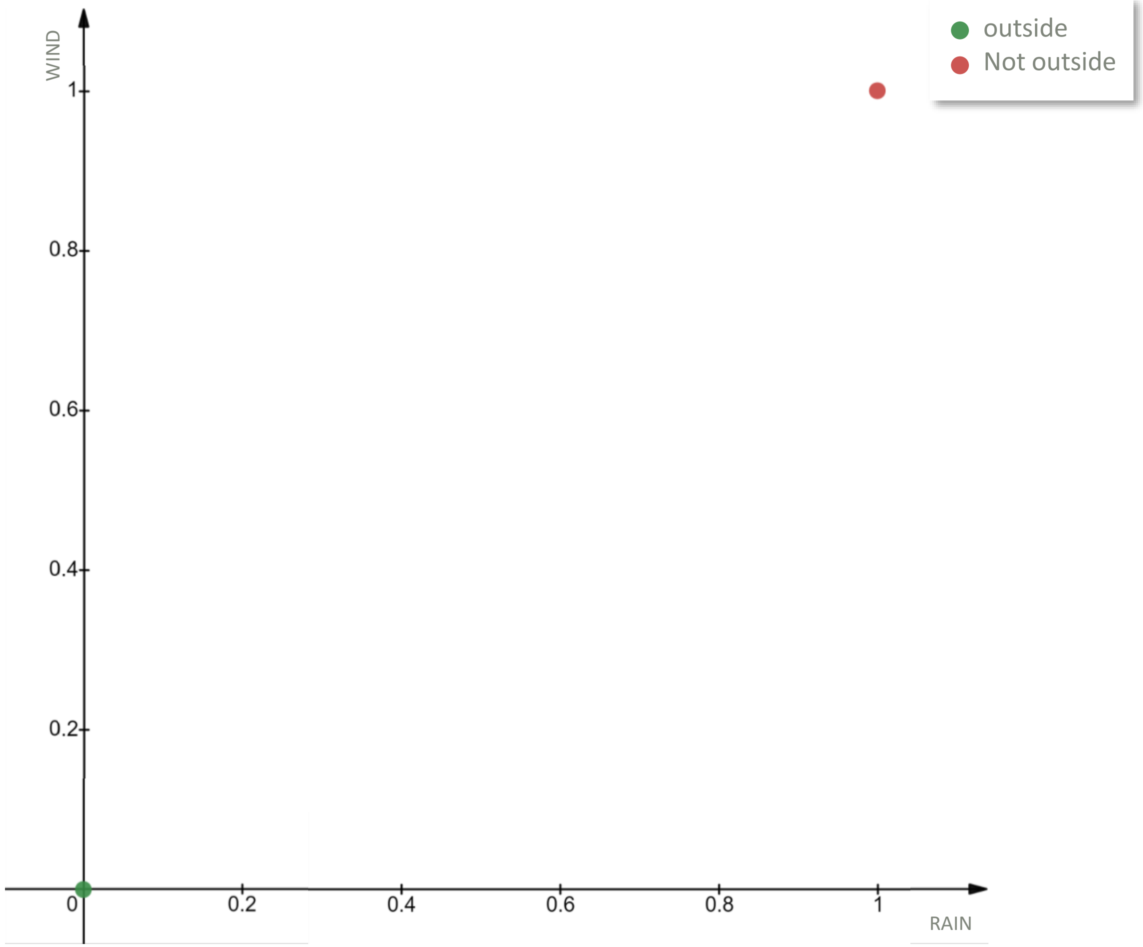
NEURAL NETWORK
MULTI-LAYER PERCEPTRON

Rain	Wind	Label
1	1	●



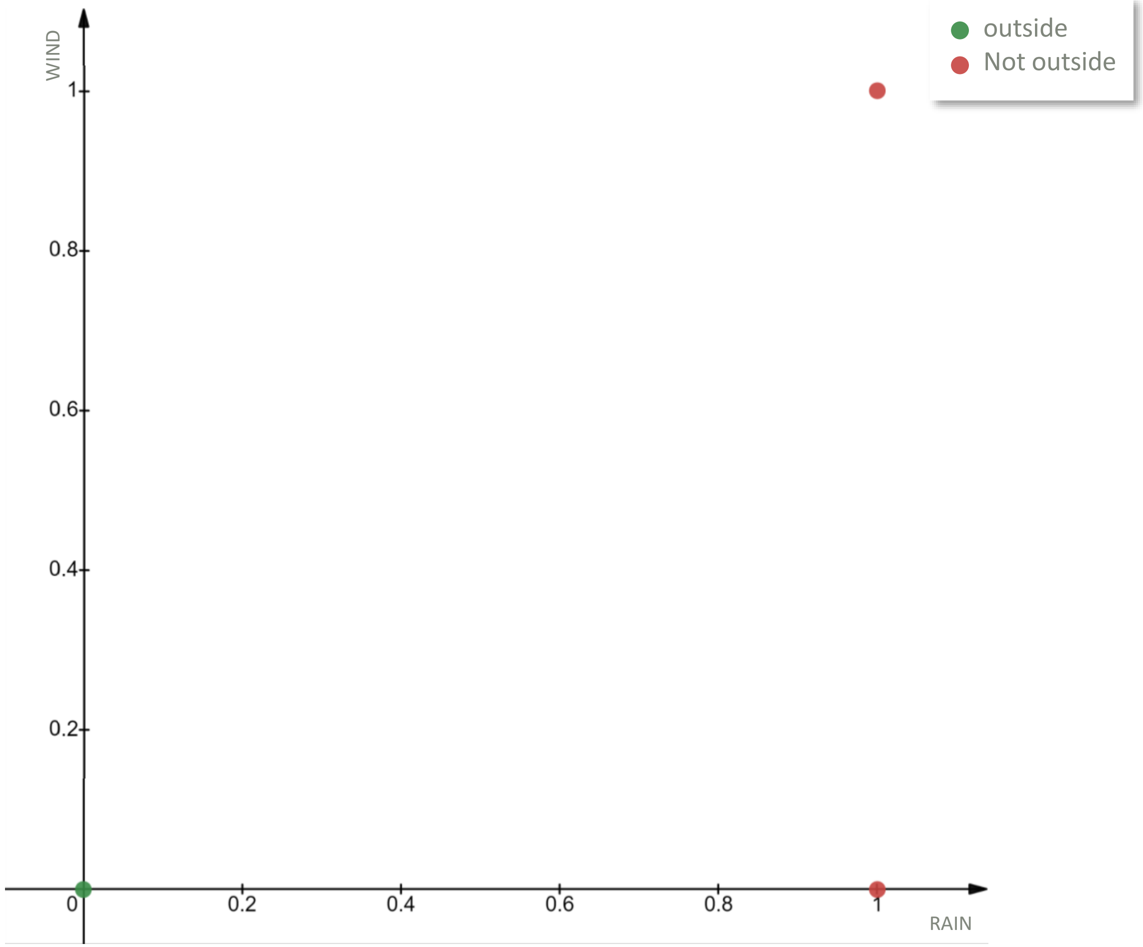
NEURAL NETWORK
MULTI-LAYER PERCEPTRON

Rain	Wind	Label
1	1	🔴
0	0	🟢



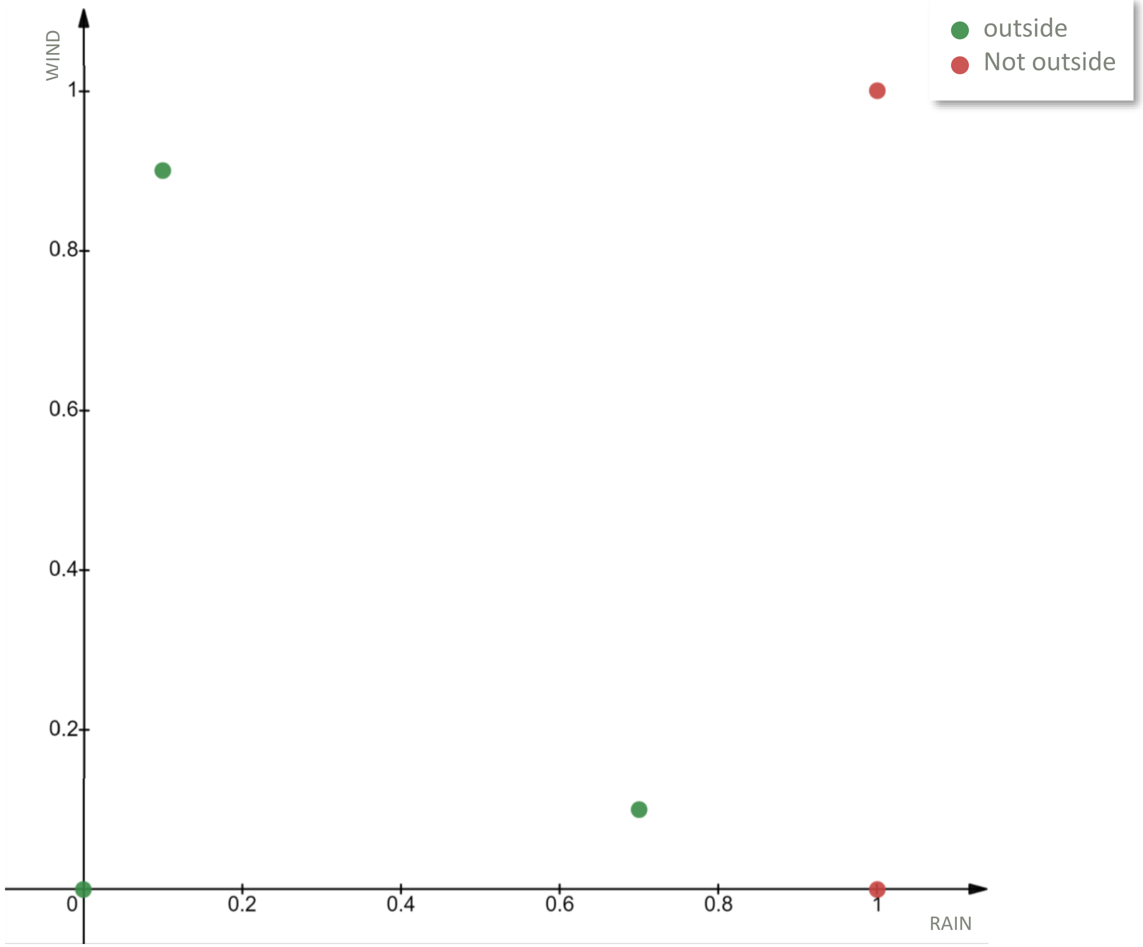
NEURAL NETWORK
MULTI-LAYER PERCEPTRON

Rain	Wind	Label
1	1	🔴
0	0	🟢
1	1	🔴



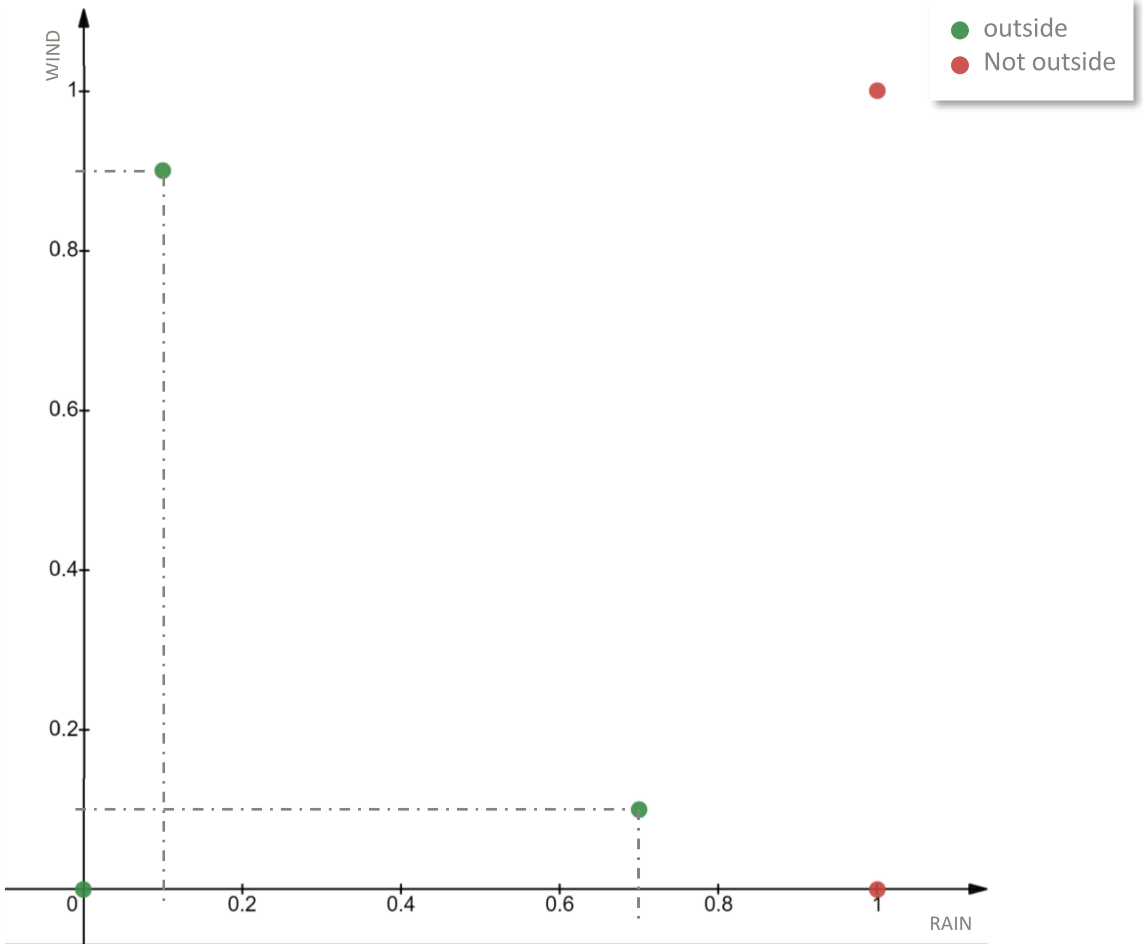
NEURAL NETWORK
MULTI-LAYER PERCEPTRON

Rain	Wind	Label
1	1	🔴
0	0	🟢
1	1	🔴
0.7	0.1	🟢
0.1	0.9	🟢



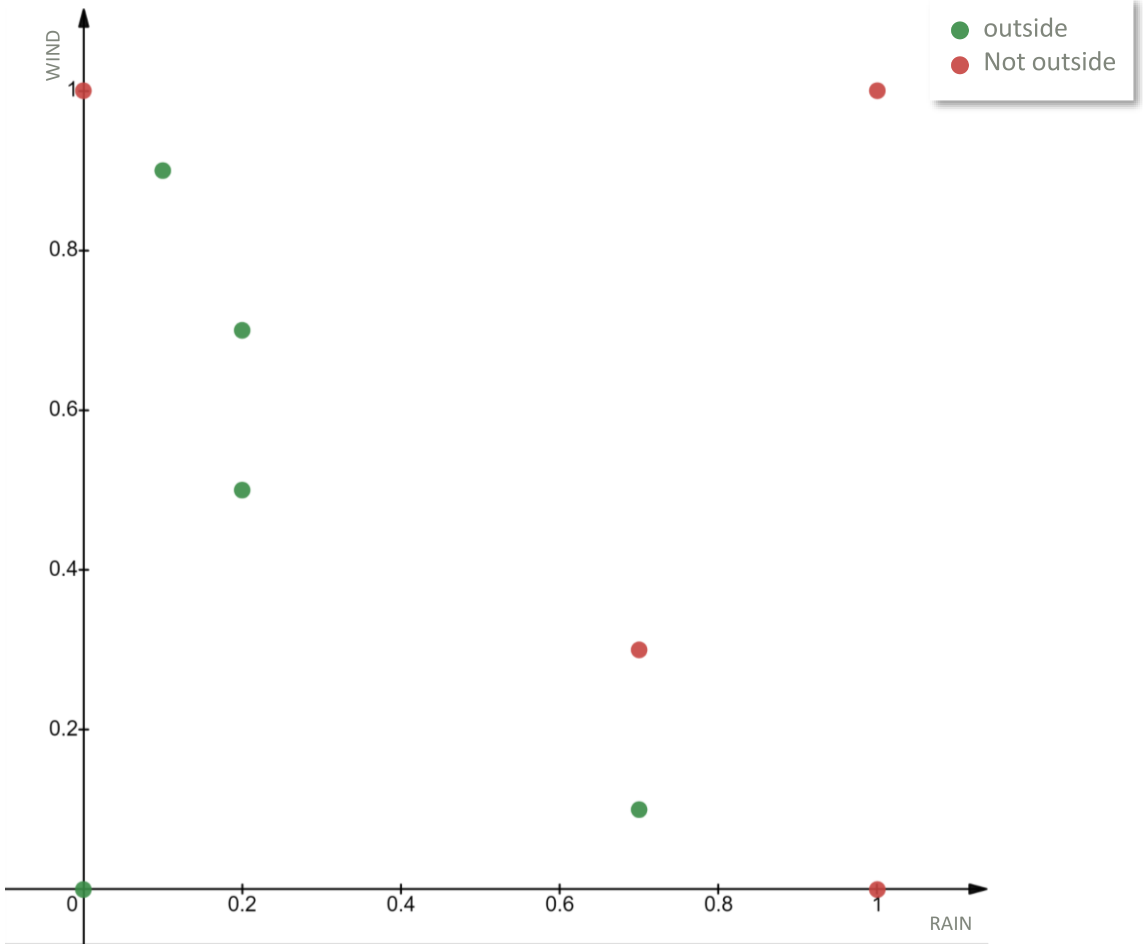
NEURAL NETWORK
MULTI-LAYER PERCEPTRON

Rain	Wind	Label
1	1	🔴
0	0	🟢
1	1	🔴
0.7	0.1	🟢
0.1	0.9	🟢



NEURAL NETWORK
MULTI-LAYER PERCEPTRON

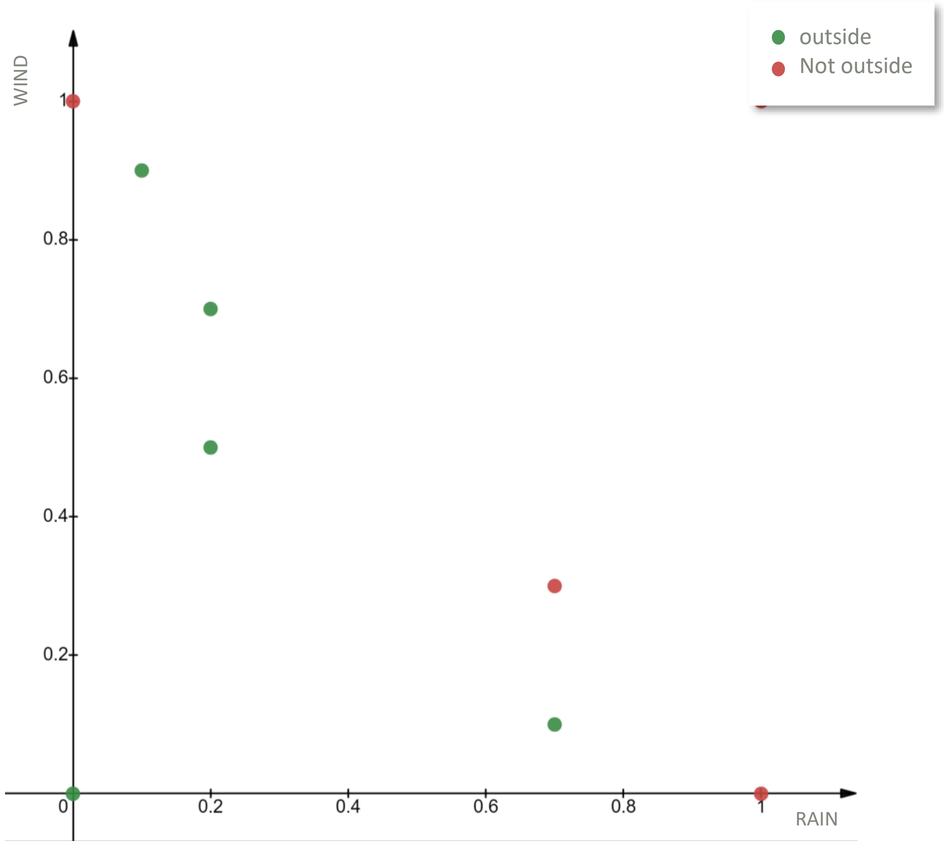
Rain	Wind	Label
1	1	🔴
0	0	🟢
1	1	🔴
0.7	0.1	🟢
0.1	0.9	🟢
0.7	0.3	🔴
0.2	0.5	🟢
0.2	0.7	🟢
0	1	🔴



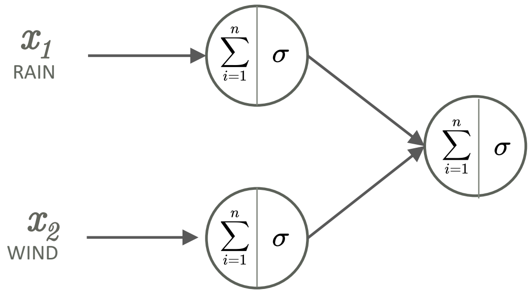
NEURAL NETWORK
MULTI-LAYER PERCEPTRON

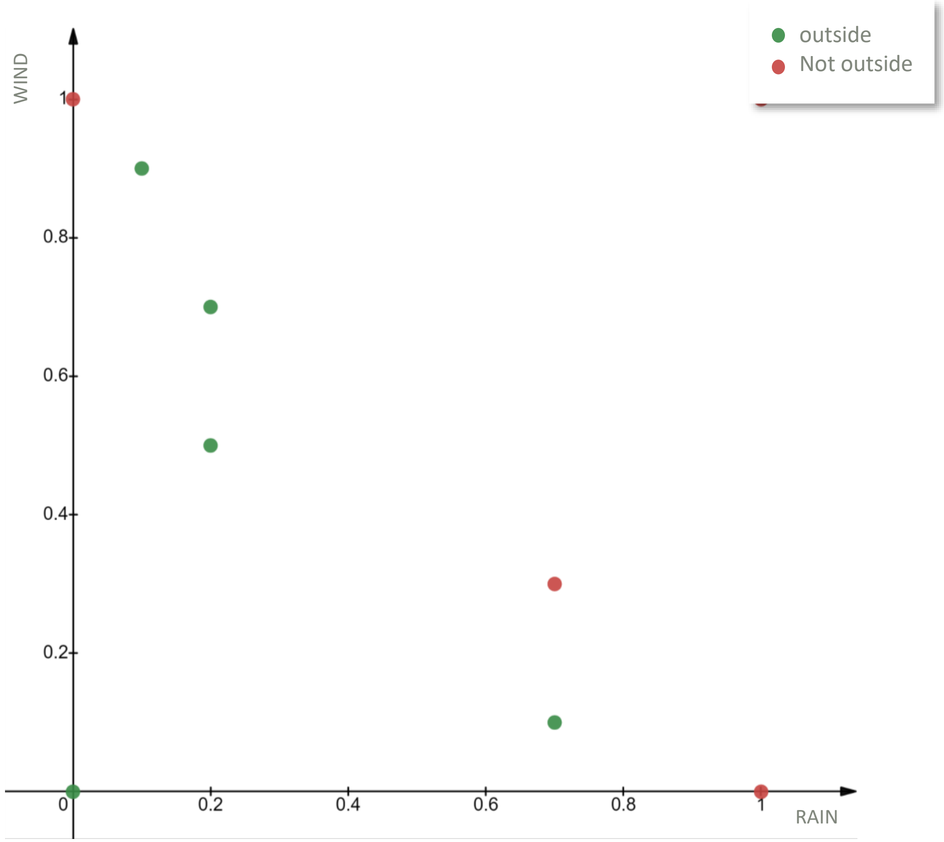
x_1
RAIN

x_2
WIND

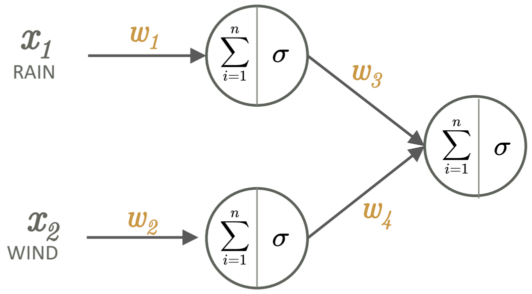


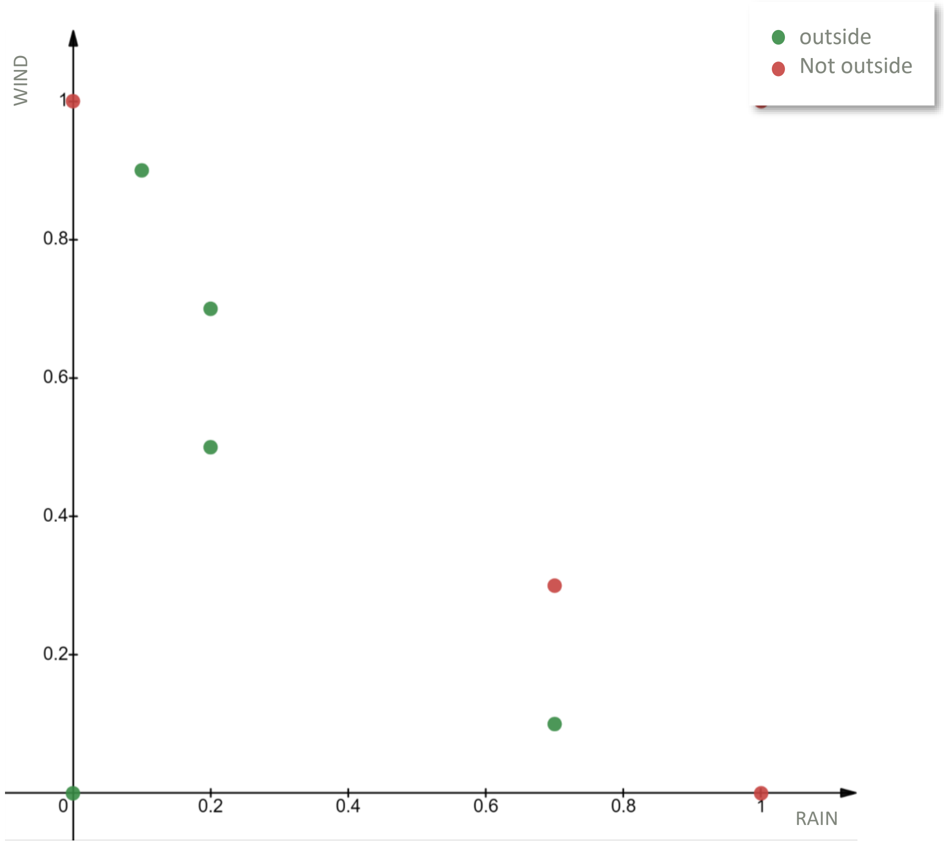
NEURAL NETWORK
MULTI-LAYER PERCEPTRON



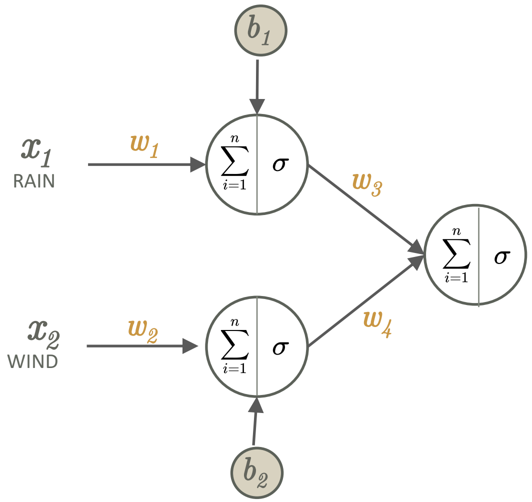


NEURAL NETWORK
MULTI-LAYER PERCEPTRON

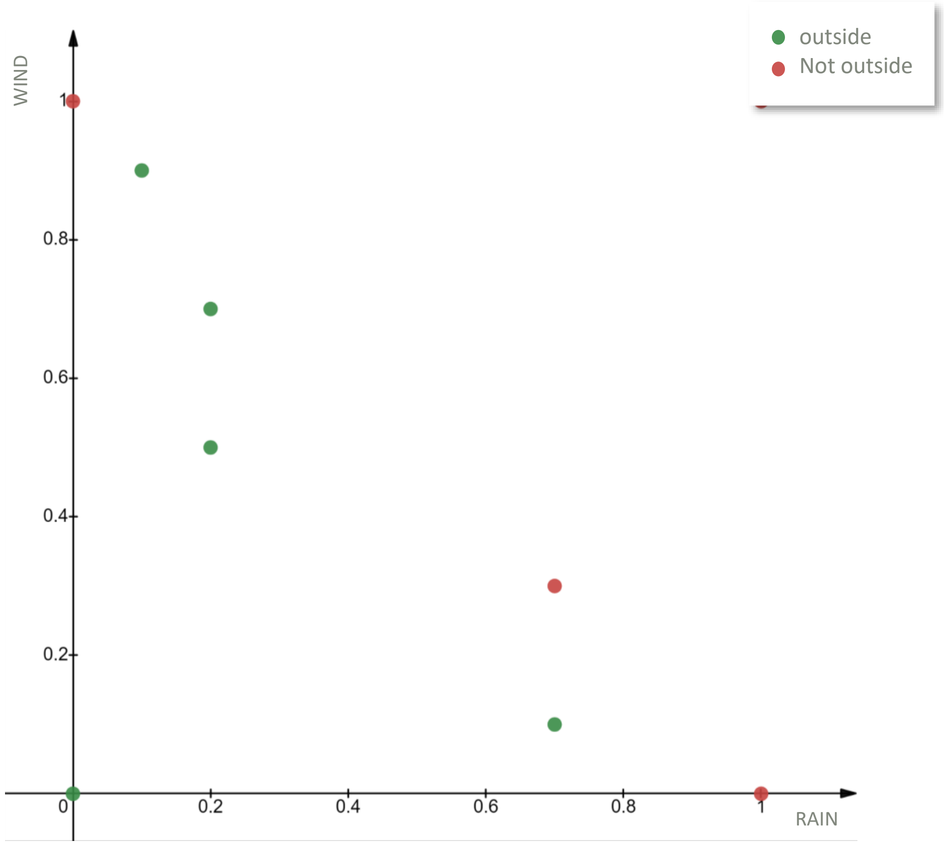




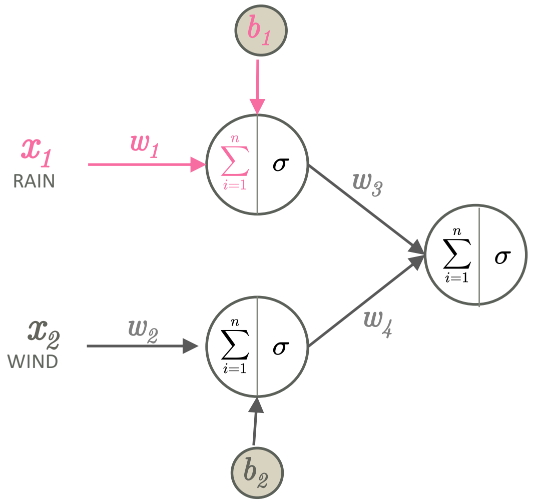
NEURAL NETWORK
MULTI-LAYER PERCEPTRON



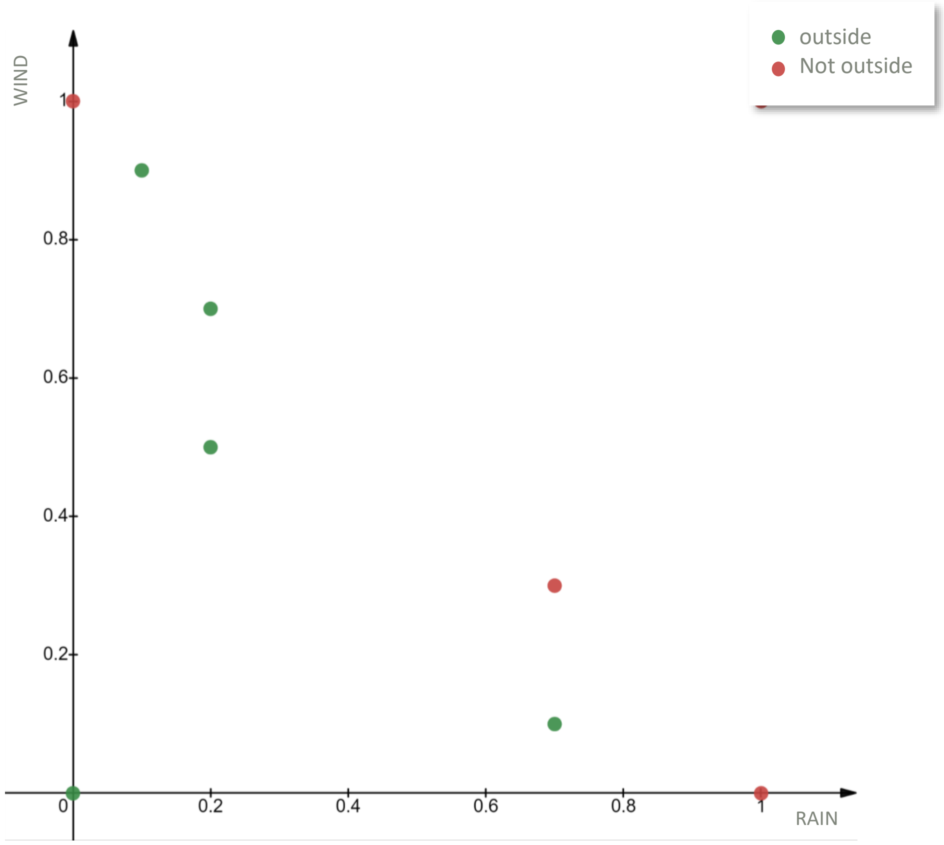
b_1
x_1
RAIN
w_1
$\sum_{i=1}^{n}$ σ
w_3
$\sum_{i=1}^{n}$ σ
x_2
WIND
w_2
$\sum_{i=1}^{n}$ σ
w_4
b_2



NEURAL NETWORK
MULTI-LAYER PERCEPTRON

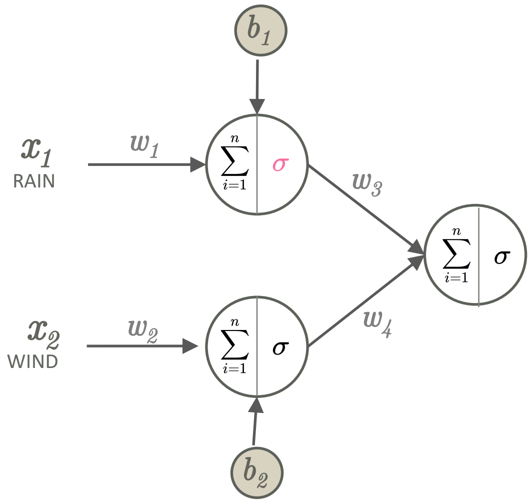


b_1
x_1
RAIN
w_1
$\sum_{i=1}^{n}$ σ
w_3
$\sum_{i=1}^{n}$ σ
x_2
WIND
w_2
$\sum_{i=1}^{n}$ σ
w_4
b_2

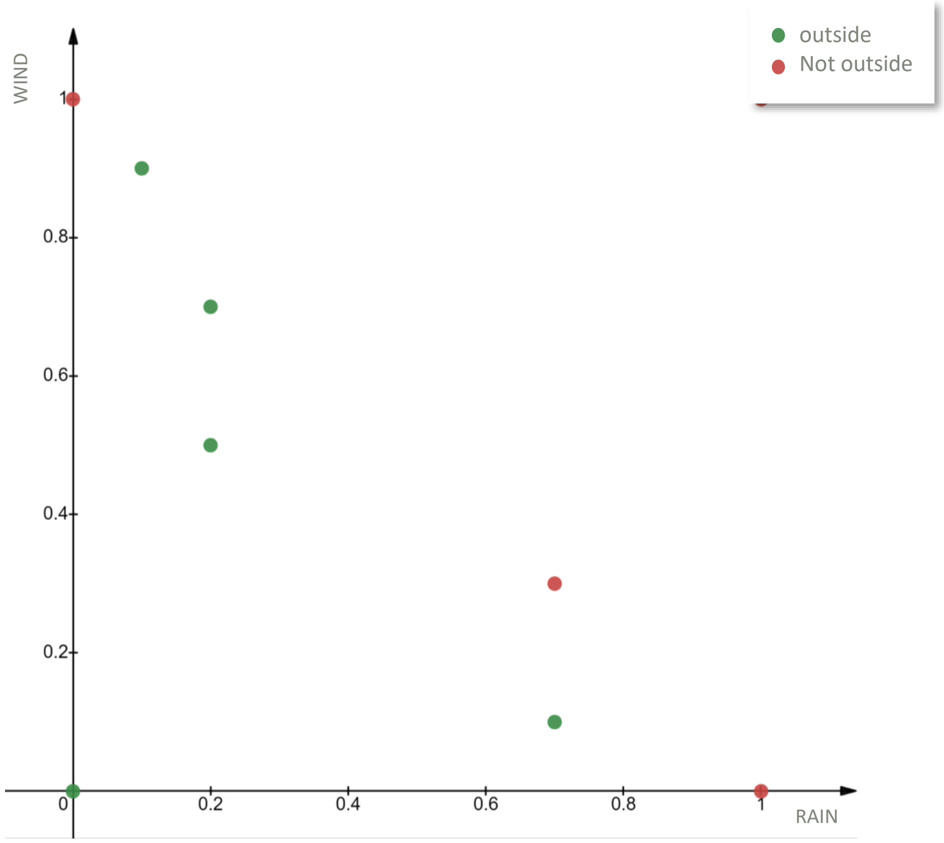


$\omega_1 * x_1 + b_1$

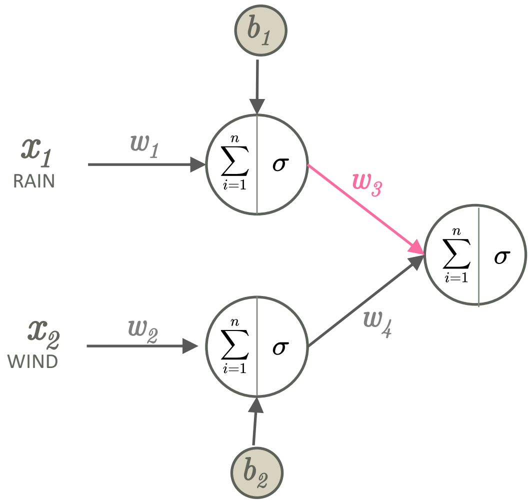
NEURAL NETWORK
MULTI-LAYER PERCEPTRON



$\sigma(\omega_1 * x_1 + b_1)$

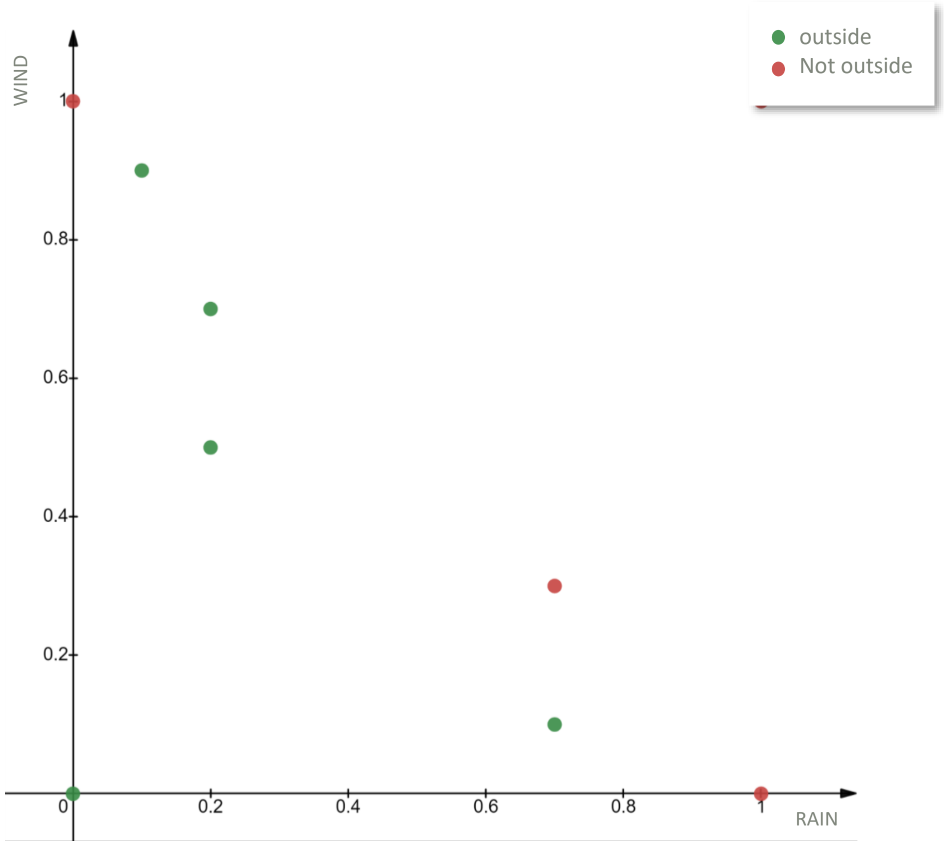


NEURAL NETWORK
MULTI-LAYER PERCEPTRON

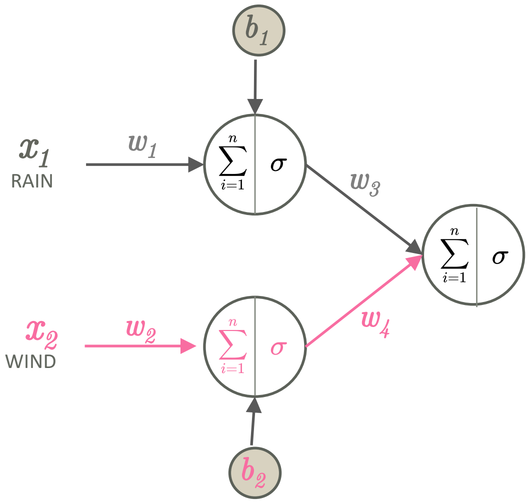


b_1
x_1
RAIN
w_1
$\sum_{i=1}^{n}$ σ
w_3
$\sum_{i=1}^{n}$ σ
x_2
WIND
w_2
$\sum_{i=1}^{n}$ σ
w_4
b_2

$$\sigma(\omega_1 * x_1 + b_1) * \omega_3$$

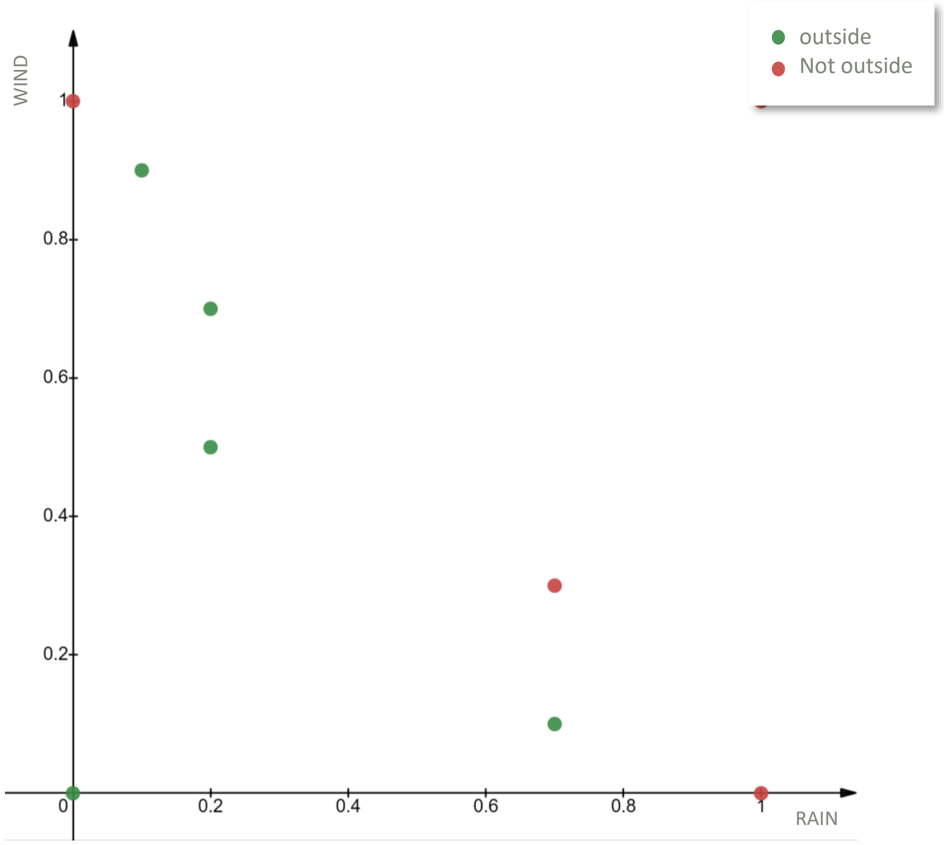


NEURAL NETWORK
MULTI-LAYER PERCEPTRON

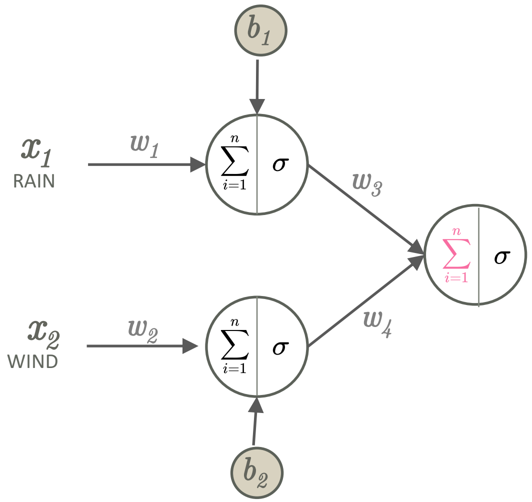


b_1
x_1
RAIN
w_1
$\sum_{i=1}^{n}$ σ
w_3
$\sum_{i=1}^{n}$ σ
x_2
WIND
w_2
$\sum_{i=1}^{n}$ σ
w_4
b_2

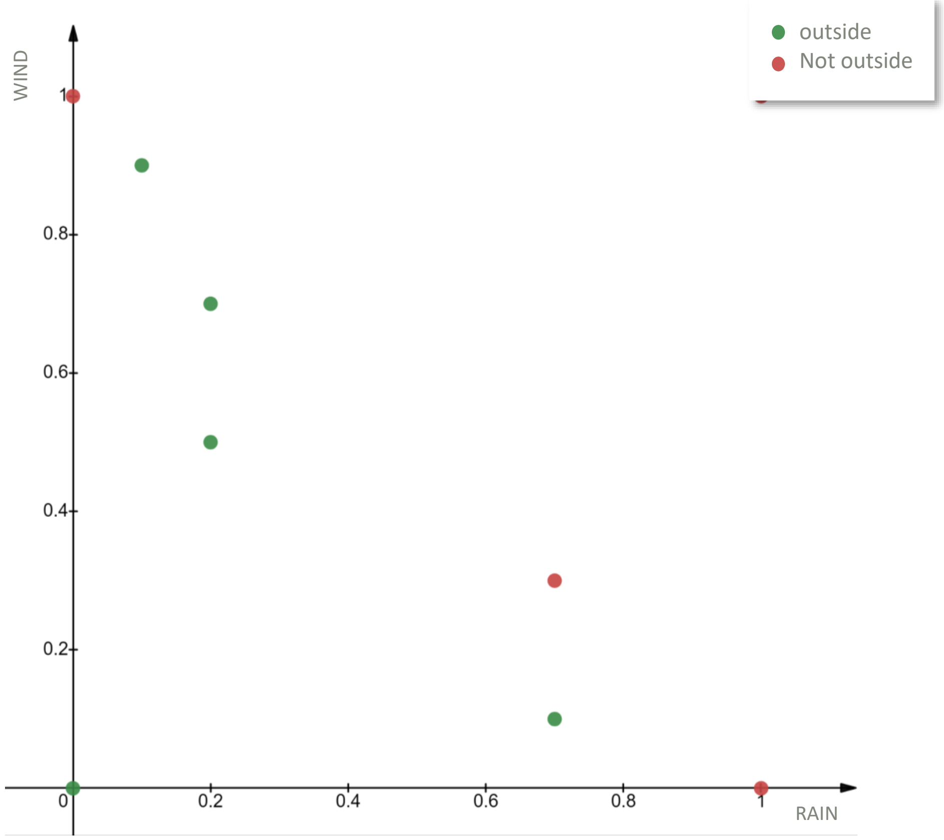
$$\sigma(\omega_1 * x_1 + b_1) * \omega_3 \quad \sigma(\omega_2 * x_2 + b_2) * \omega_4$$



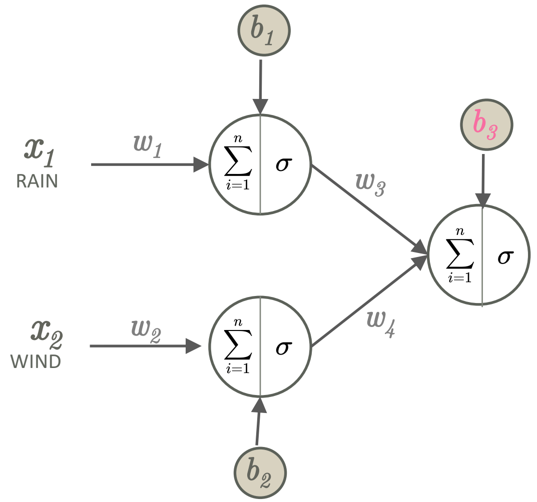
NEURAL NETWORK
MULTI-LAYER PERCEPTRON



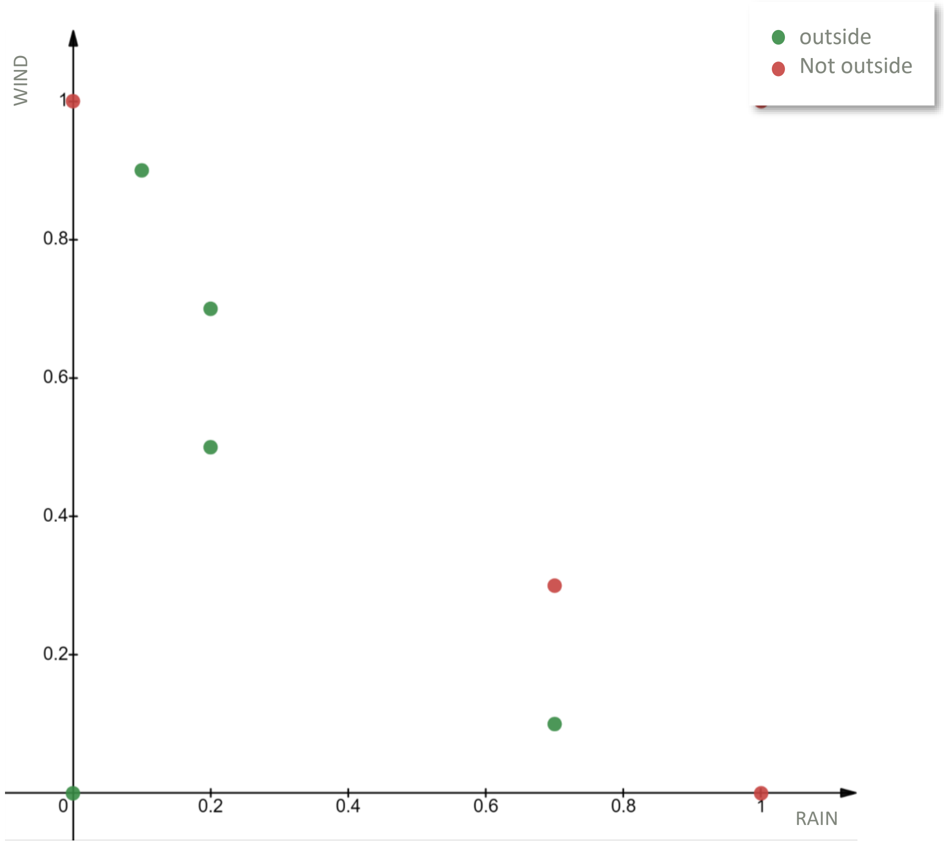
$$\sigma(\omega_1 * x_1 + b_1) * \omega_3 + \sigma(\omega_2 * x_2 + b_2) * \omega_4$$



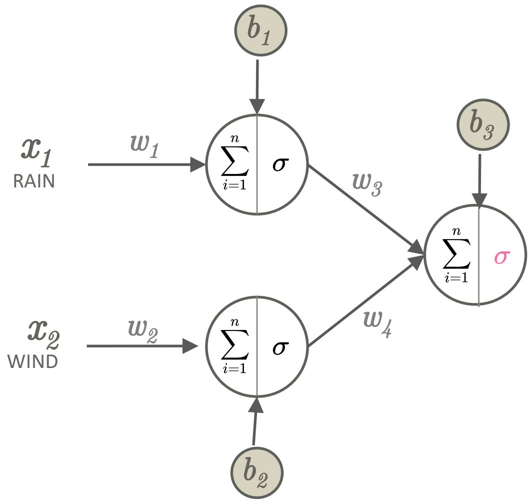
NEURAL NETWORK
MULTI-LAYER PERCEPTRON


b_1
x_1
RAIN
w_1
$\sum_{i=1}^{n}$ σ
b_3
w_3
$\sum_{i=1}^{n}$ σ
x_2
WIND
w_2
$\sum_{i=1}^{n}$ σ
w_4
b_2

$$\sigma(\omega_1 * x_1 + b_1) * \omega_3 + \sigma(\omega_2 * x_2 + b_2) * \omega_4 + b_3$$

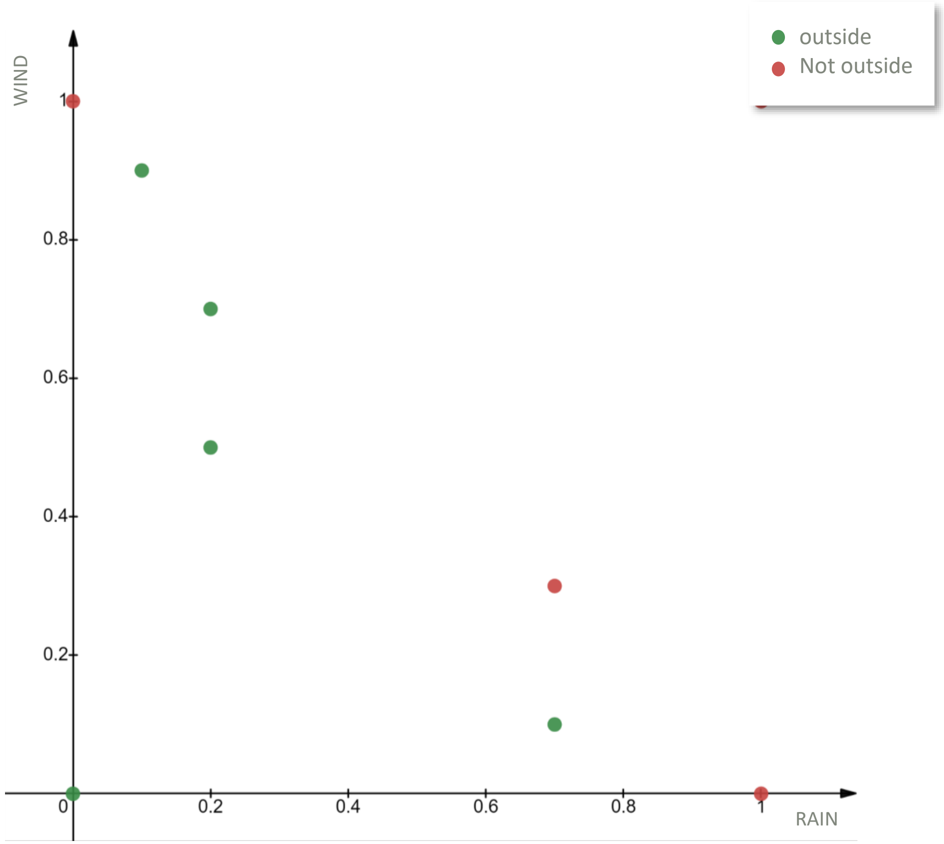


NEURAL NETWORK
MULTI-LAYER PERCEPTRON

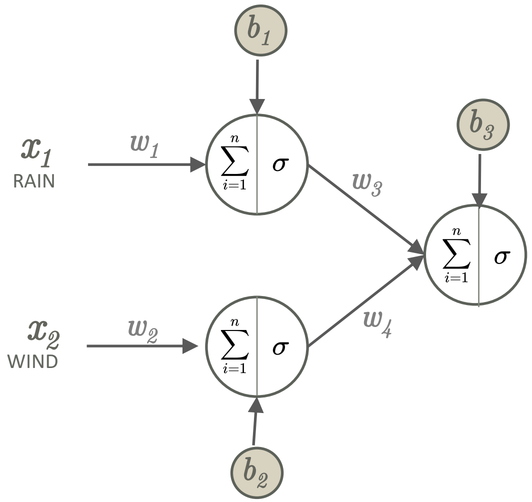


b_1
x_1
RAIN
w_1
$\sum_{i=1}^{n}$ σ
w_3
b_3
$\sum_{i=1}^{n}$ σ
x_2
WIND
w_2
$\sum_{i=1}^{n}$ σ
w_4
b_2

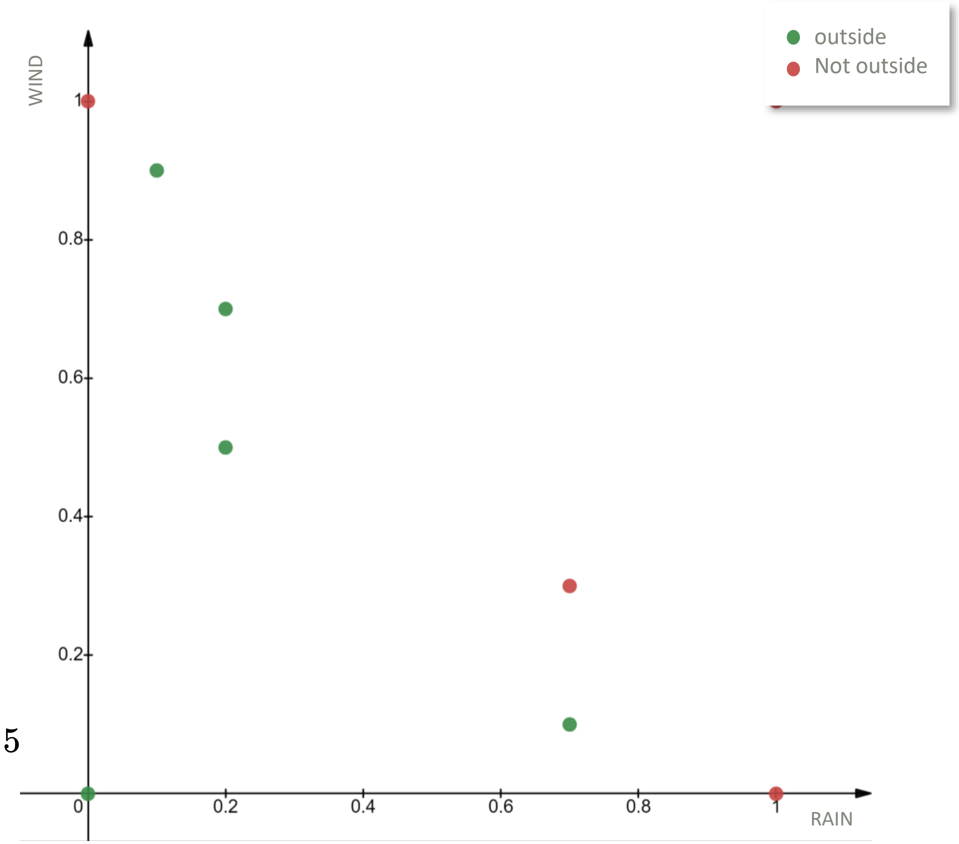
$$\sigma\left(\sigma(\omega_1 * x_1 + b_1) * \omega_3 + \sigma(\omega_2 * x_2 + b_2) * \omega_4 + b_3\right)$$



NEURAL NETWORK
MULTI-LAYER PERCEPTRON

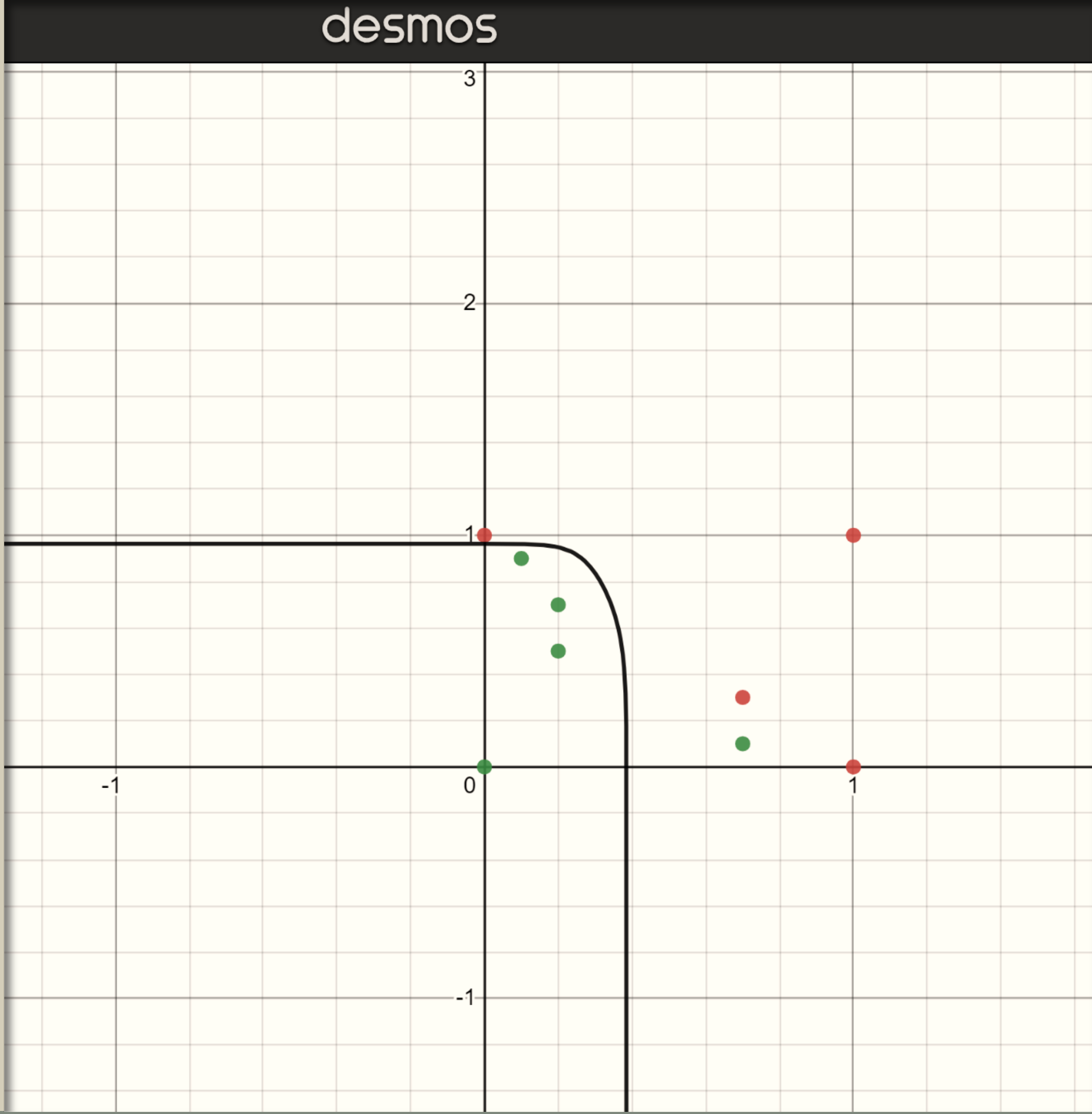


$$\sigma\left(\sigma(\omega_1 * x_1 + b_1) * \omega_3 + \sigma(\omega_2 * x_2 + b_2) * \omega_4 + b_3\right) = 0.5$$

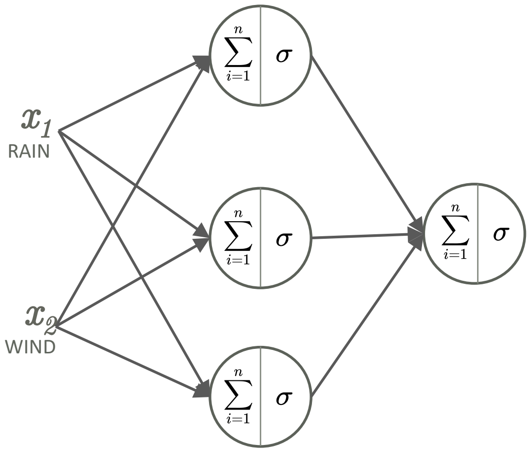


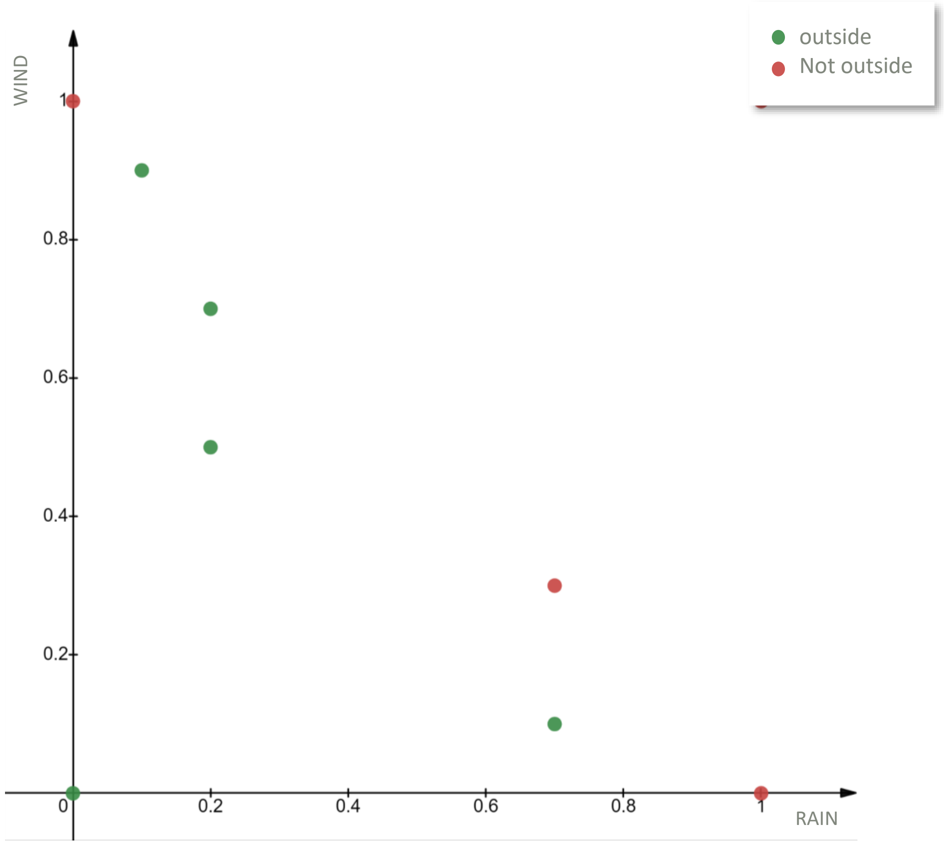
TIME FOR
DESMOS

https://www.desmos.com/calculator/1czmp32hjs

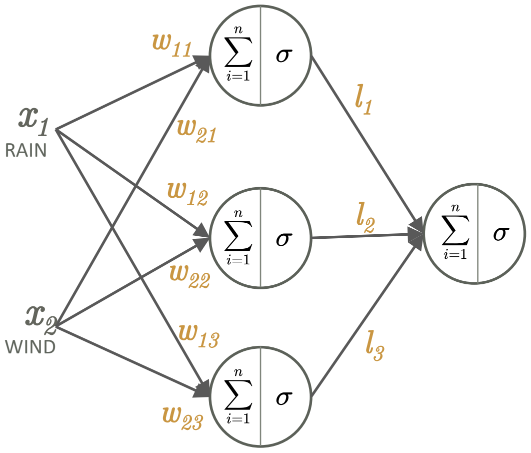


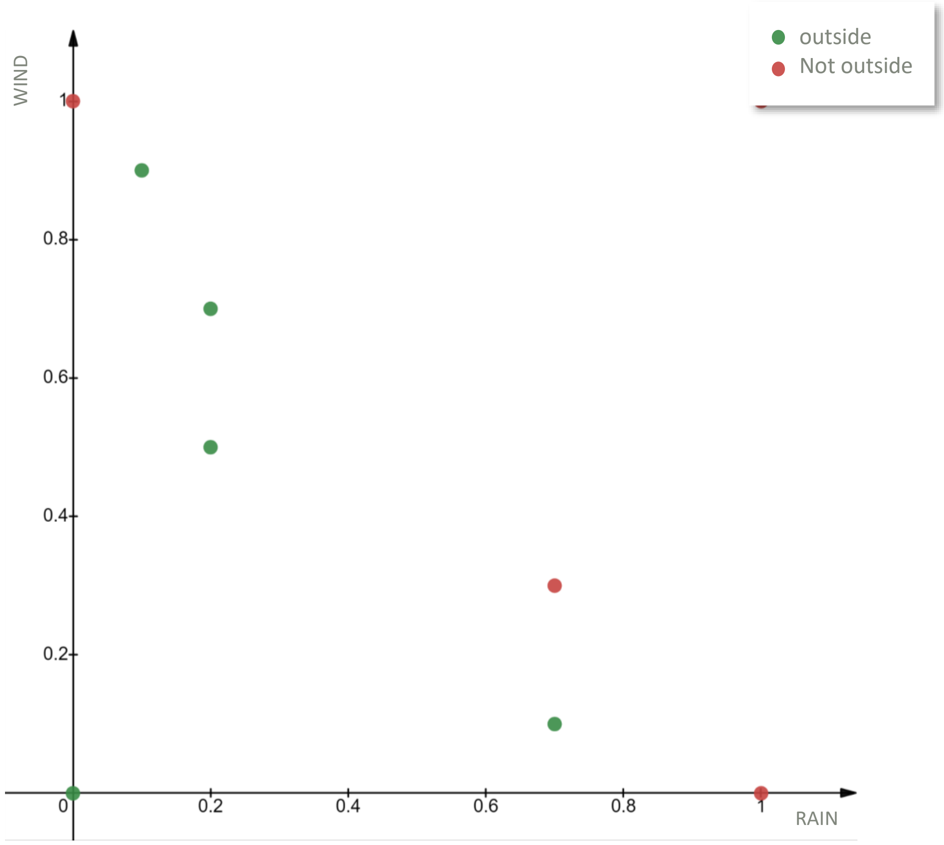
NEURAL NETWORK
MULTI-LAYER PERCEPTRON


x_1
RAIN
x_2
WIND
$\sum_{i=1}^{n}$ σ
$\sum_{i=1}^{n}$ σ
$\sum_{i=1}^{n}$ σ
$\sum_{i=1}^{n}$ σ

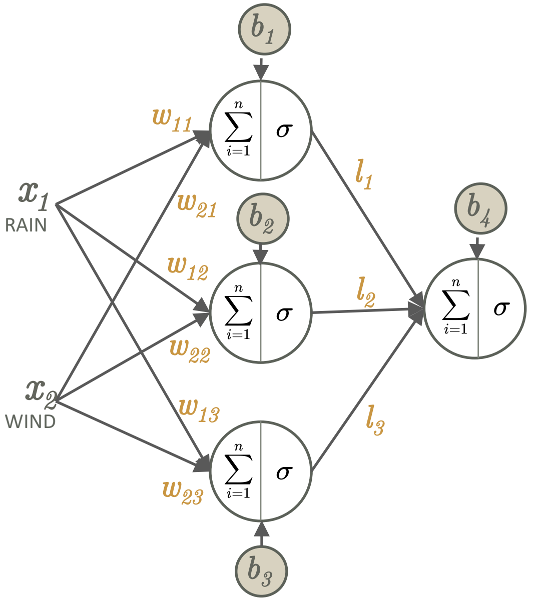


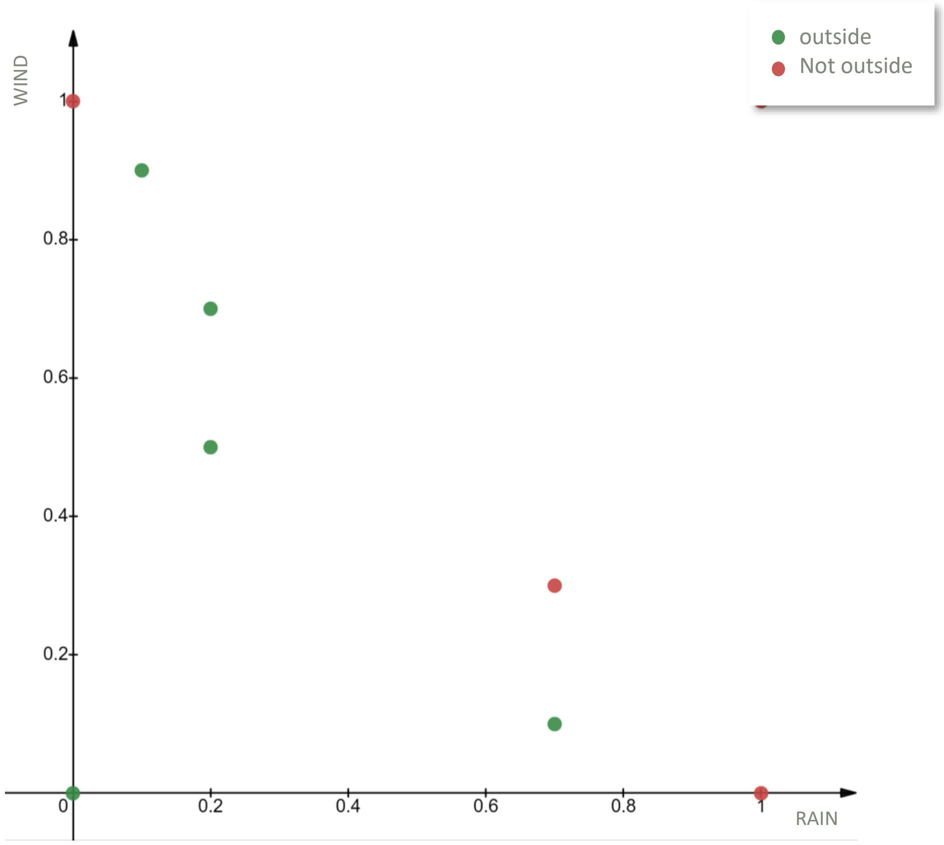
NEURAL NETWORK
MULTI-LAYER PERCEPTRON



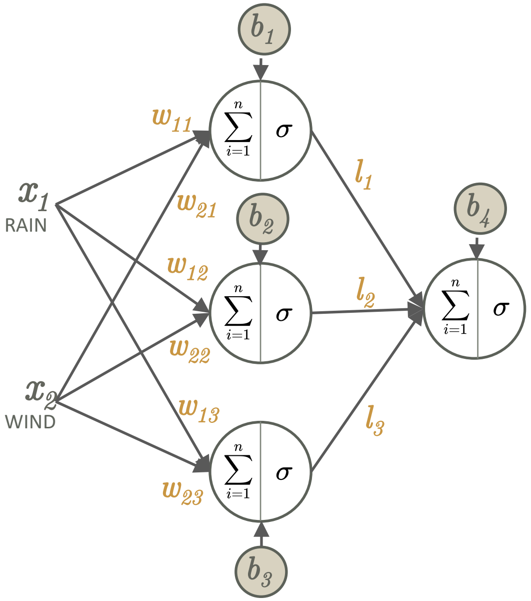


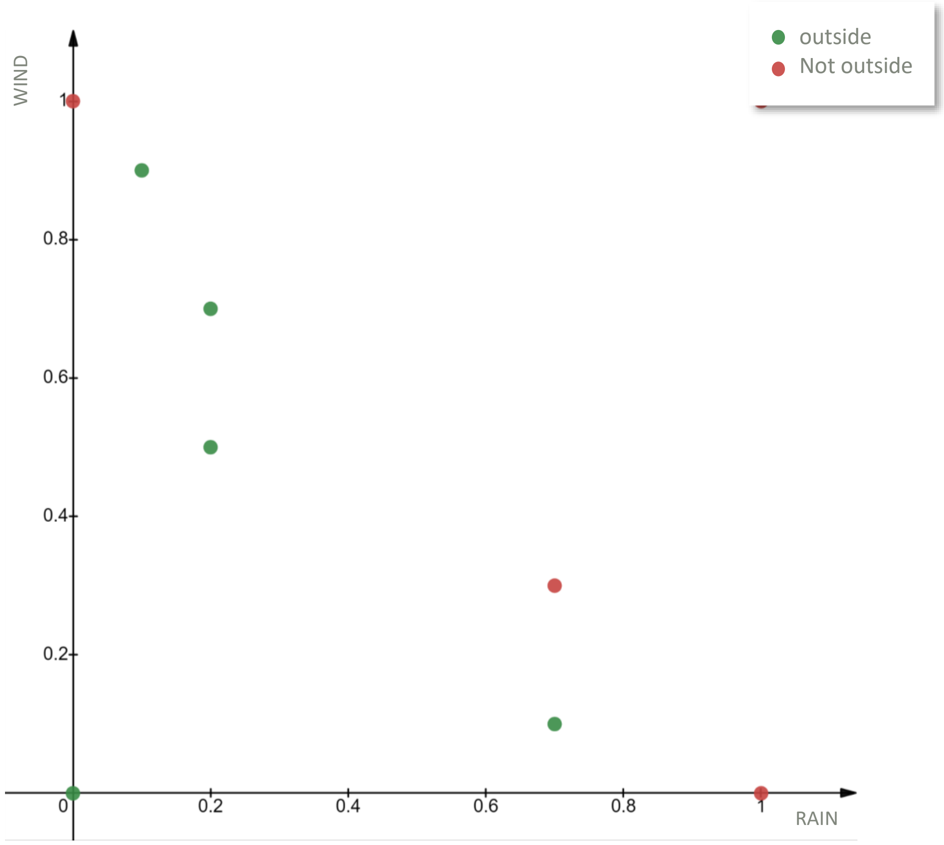
NEURAL NETWORK
MULTI-LAYER PERCEPTRON





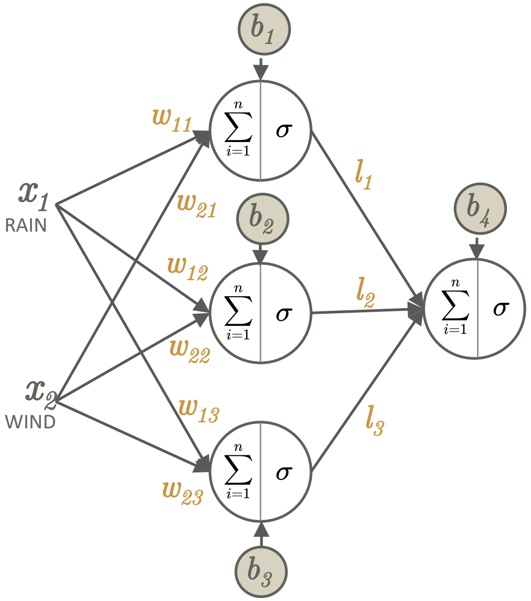
NEURAL NETWORK
MULTI-LAYER PERCEPTRON

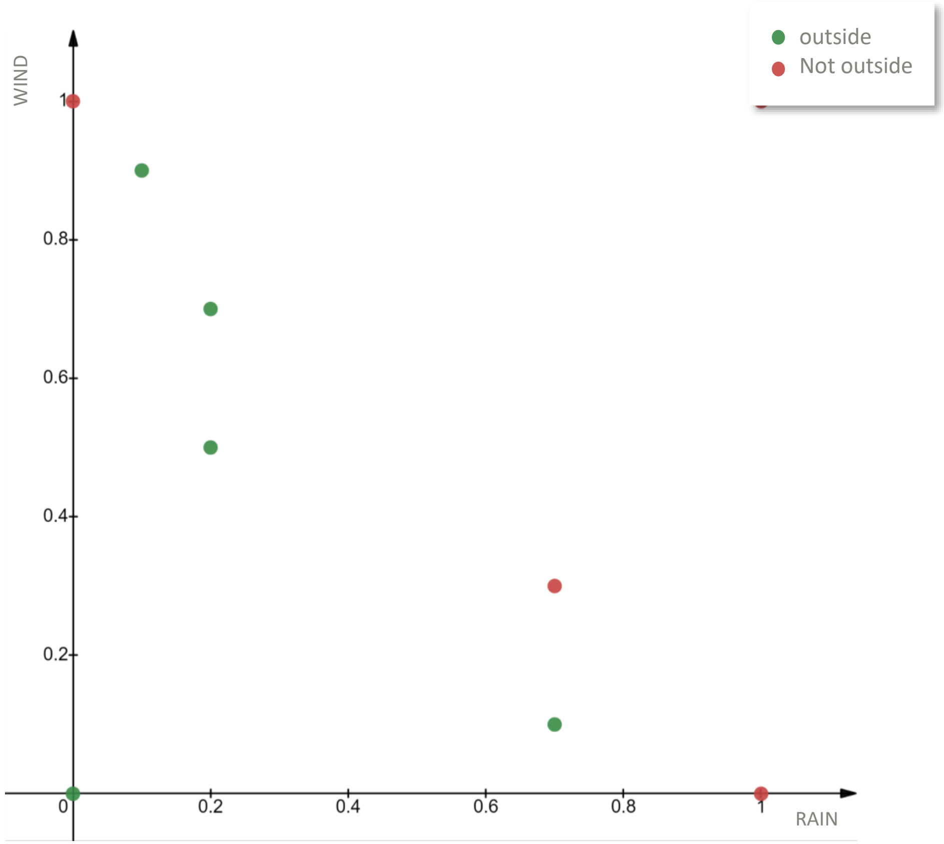




$$\sigma\left(\sigma(b_1+\sum_{i=1}^{n}\omega_{i1}*x_i)*l_1+\sigma(b_2+\sum_{i=1}^{n}\omega_{i2}*x_i)*l_2+\sigma(b_3+\sum_{i=1}^{n}\omega_{i3}*x_i)*l_3+b_4\right)=0.5$$

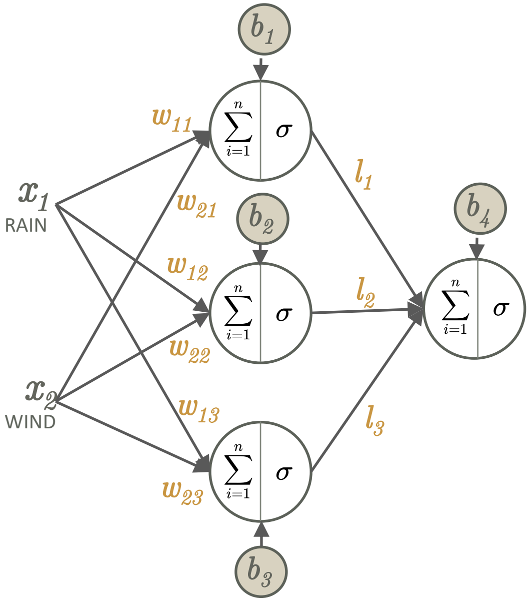
NEURAL NETWORK
MULTI-LAYER PERCEPTRON

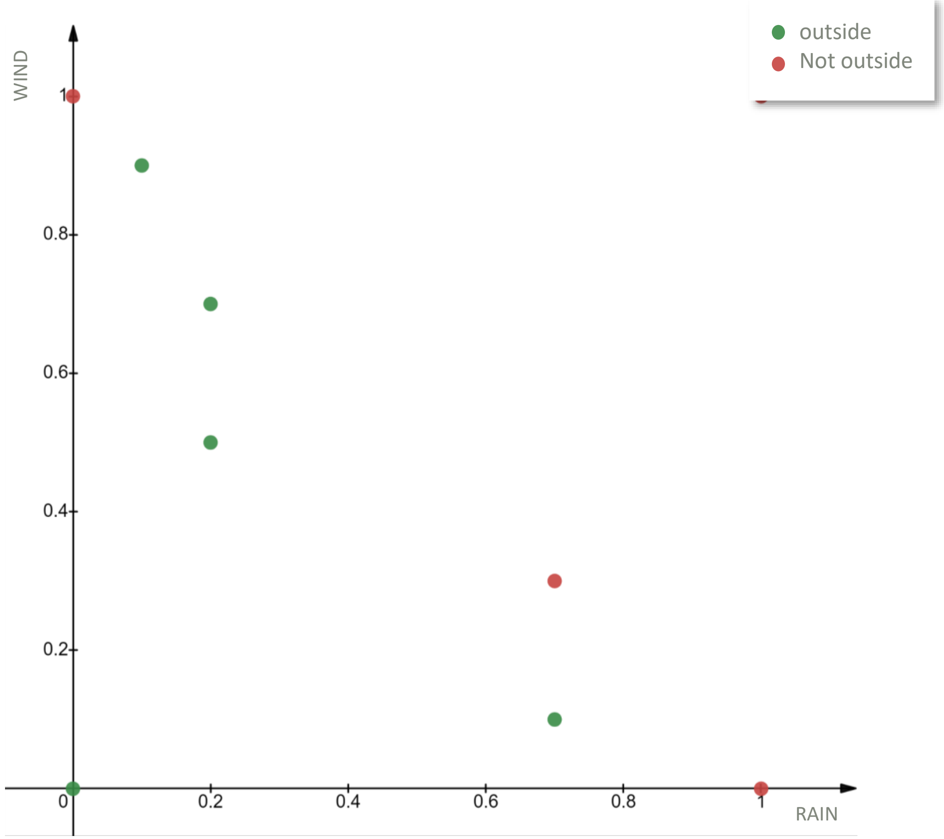




$$\sigma\left(\sigma(b_1+\omega_{11}*x_1+\omega_{21}*x_2)*l_1+\sigma(b_2+\omega_{12}*x_1+\omega_{22}*x_2)*l_2+\sigma(b_3+\omega_{13}*x_1+\omega_{23}*x_2)*l_3+b_4\right)=0.5$$

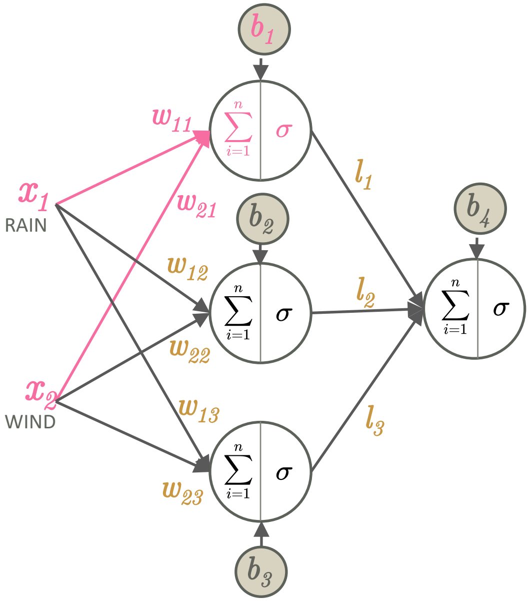
NEURAL NETWORK
MULTI-LAYER PERCEPTRON

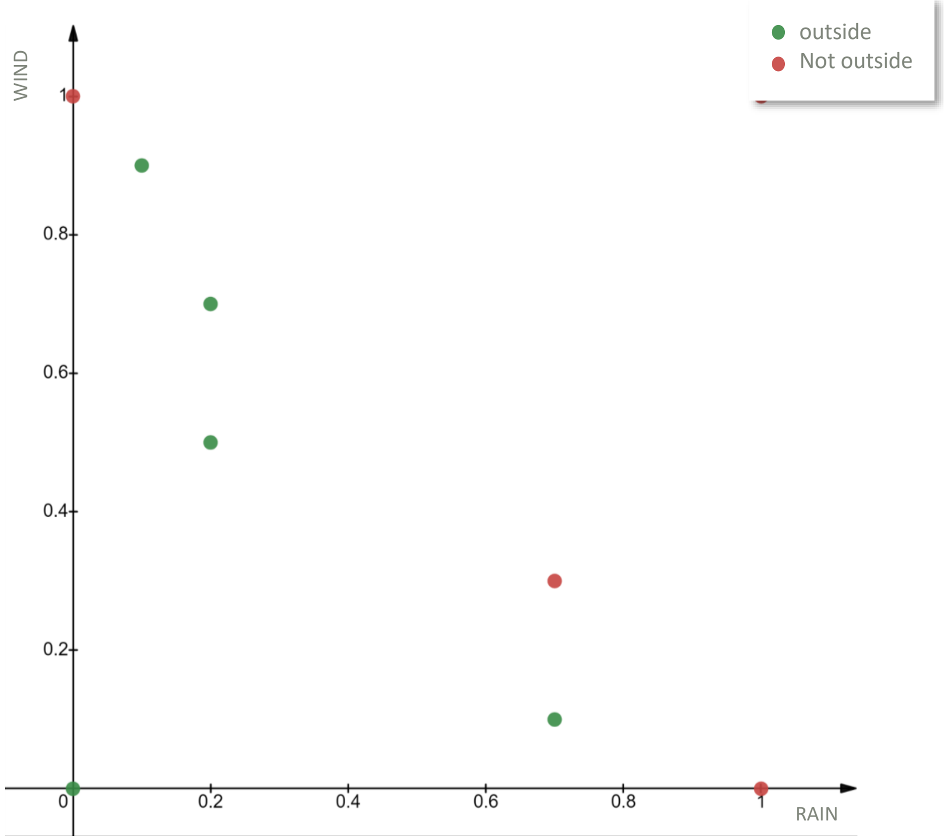




$$\sigma\left(\sigma(b_1+\omega_{11}*x_1+\omega_{21}*x_2)*l_1+\sigma(b_2+\omega_{12}*x_1+\omega_{22}*x_2)*l_2+\sigma(b_3+\omega_{13}*x_1+\omega_{23}*x_2)*l_3+b_4\right)=0.5$$

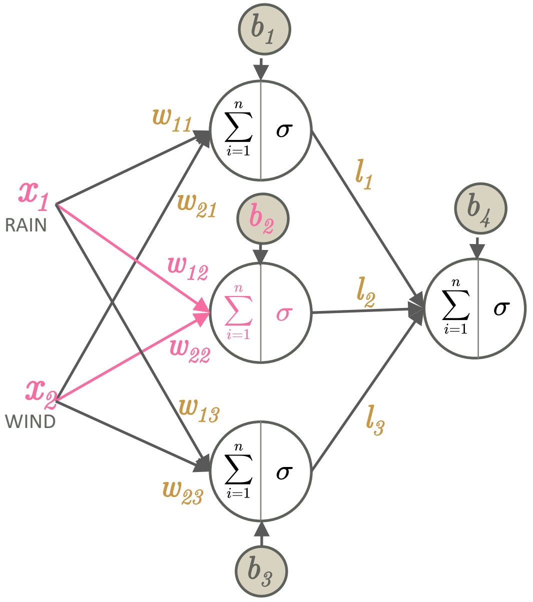
NEURAL NETWORK
MULTI-LAYER PERCEPTRON

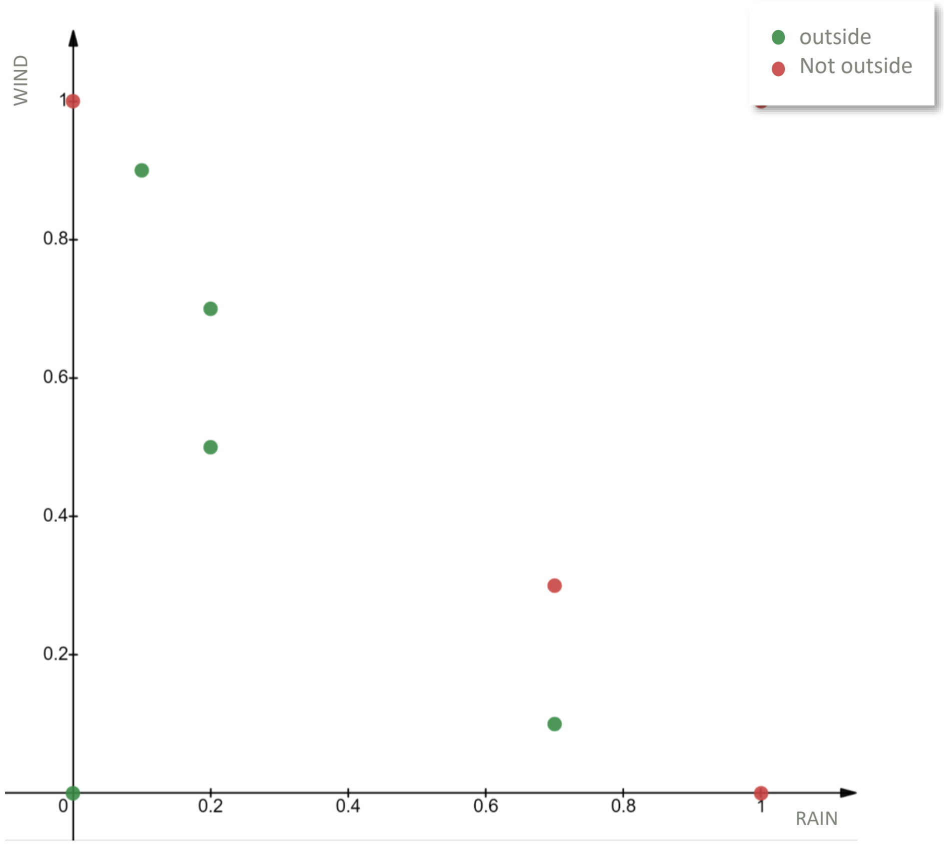




$$\sigma\left(\sigma(b_1+\omega_{11}*x_1+\omega_{21}*x_2)*l_1+\sigma(b_2+\omega_{12}*x_1+\omega_{22}*x_2)*l_2+\sigma(b_3+\omega_{13}*x_1+\omega_{23}*x_2)*l_3+b_4\right)=0.5$$

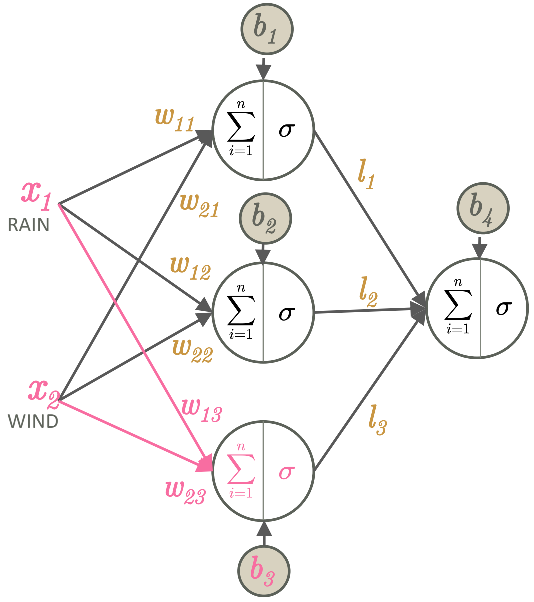
NEURAL NETWORK
MULTI-LAYER PERCEPTRON

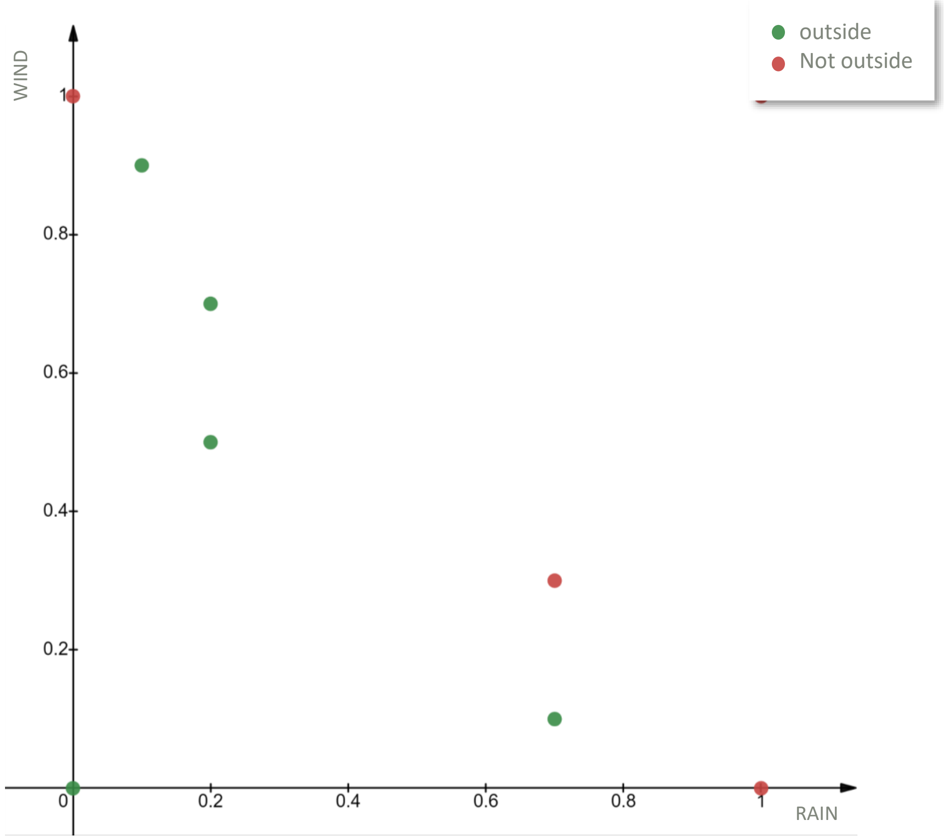




$$\sigma\left(\sigma(b_1+\omega_{11}*x_1+\omega_{21}*x_2)*l_1+\sigma(b_2+\omega_{12}*x_1+\omega_{22}*x_2)*l_2+\sigma(b_3+\omega_{13}*x_1+\omega_{23}*x_2)*l_3+b_4\right)=0.5$$

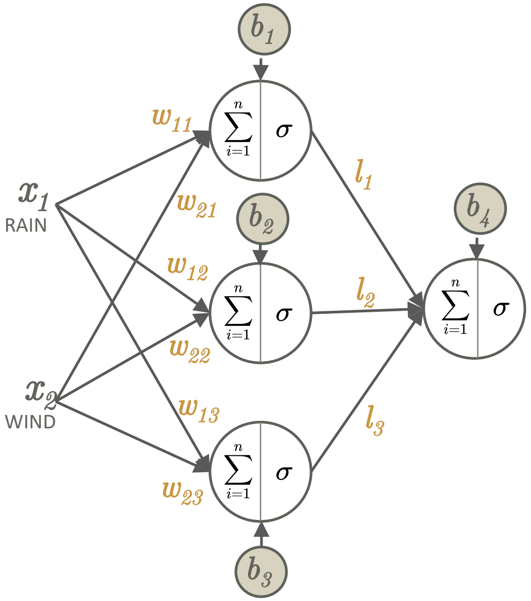
NEURAL NETWORK
MULTI-LAYER PERCEPTRON

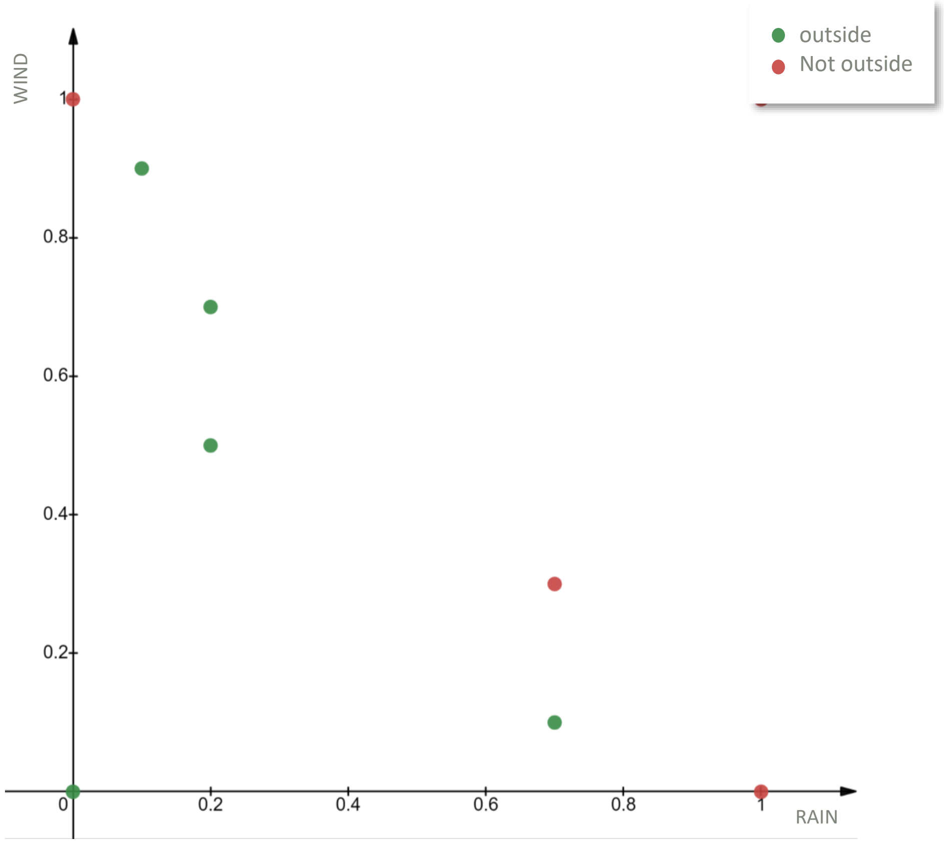




$$\sigma\left(\sigma(b_1 + \omega_{11} * x_1 + \omega_{21} * x_2) * l_1 + \sigma(b_2 + \omega_{12} * x_1 + \omega_{22} * x_2) * l_2 + \sigma(b_3 + \omega_{13} * x_1 + \omega_{23} * x_2) * l_3 + b_4\right) = 0.5$$

NEURAL NETWORK
MULTI-LAYER PERCEPTRON

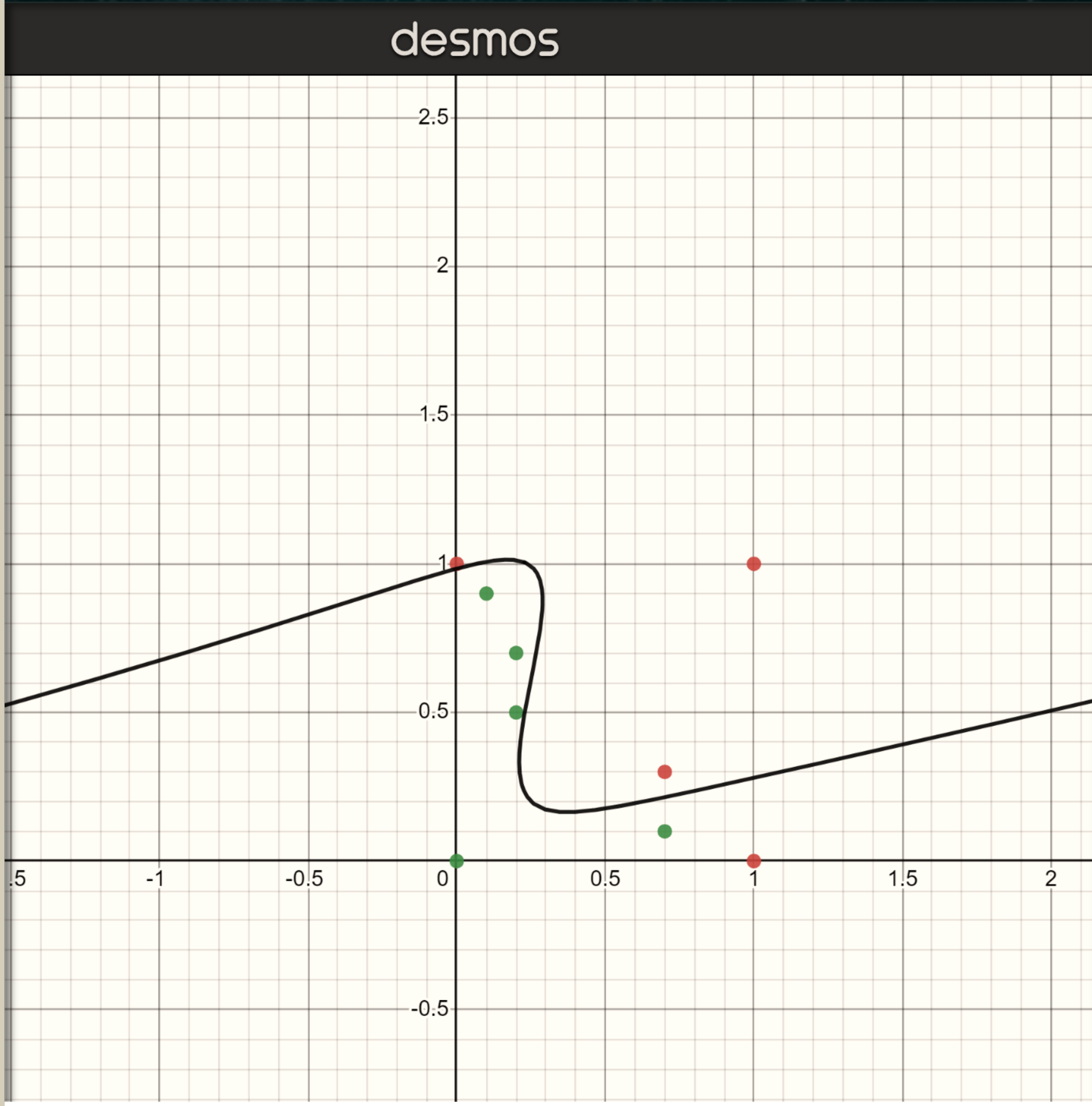




$$\sigma\left(\sigma(b_1+\omega_{11}*x_1+\omega_{21}*x_2)*l_1+\sigma(b_2+\omega_{12}*x_1+\omega_{22}*x_2)*l_2+\sigma(b_3+\omega_{13}*x_1+\omega_{23}*x_2)*l_3+b_4\right)=0.5$$

TIME FOR DESMOS

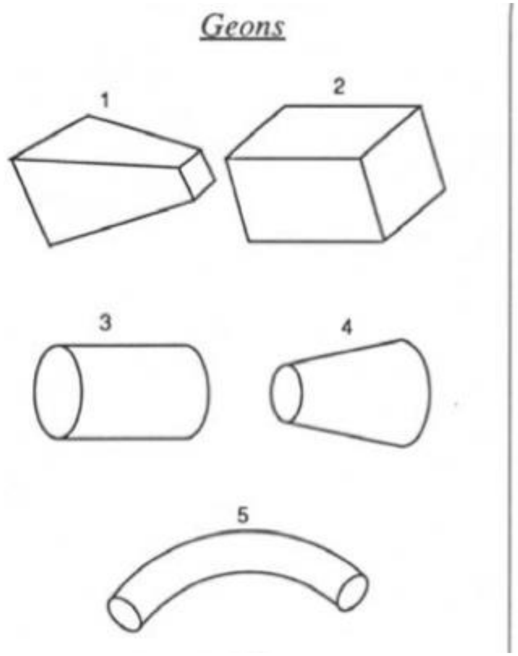
https://www.desmos.com/calculator/4dtlyi0bjg?



INTRO
COGNITIVE PSYCHOLOGY

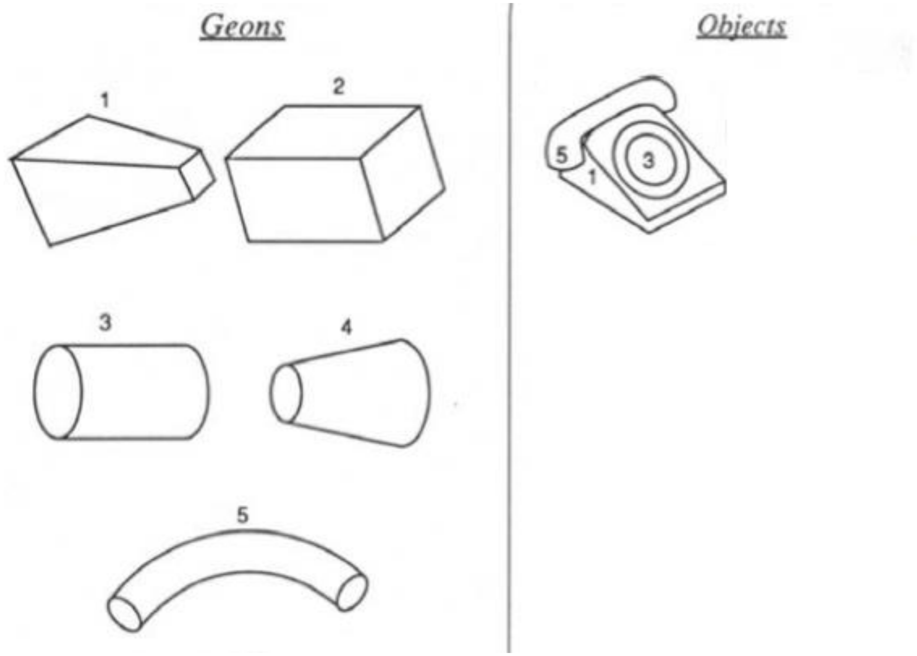
GEONS AND BIGRAM DETECTORS

- GEONS (geometry icons) are simple shapes
- From the network theory known as **recognition by components (RPC model)** (Hummel & Biederman 1992)
- According to Biederman (1987, 1880), we need (at most) 36 different GEONS to describe every object.



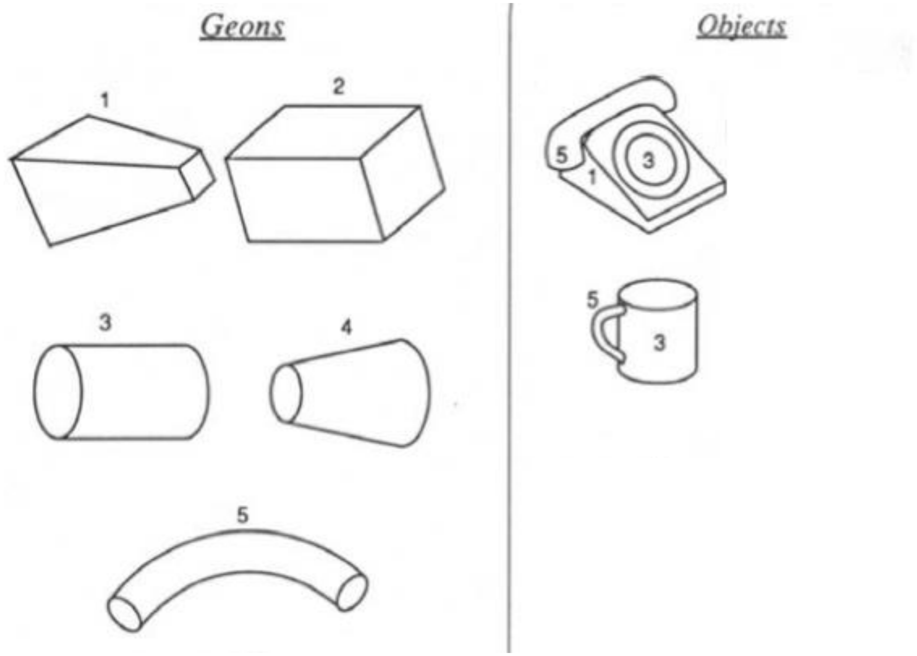
GEONS AND
BIGRAM DETECTORS

- GEONS (geometry icons) are simple shapes
- From the network theory known as **recognition by components (RPC model)** (Hummel & Biederman 1992)
- According to Biederman (1987, 1880), we need (at most) 36 different GEONS to describe every object.



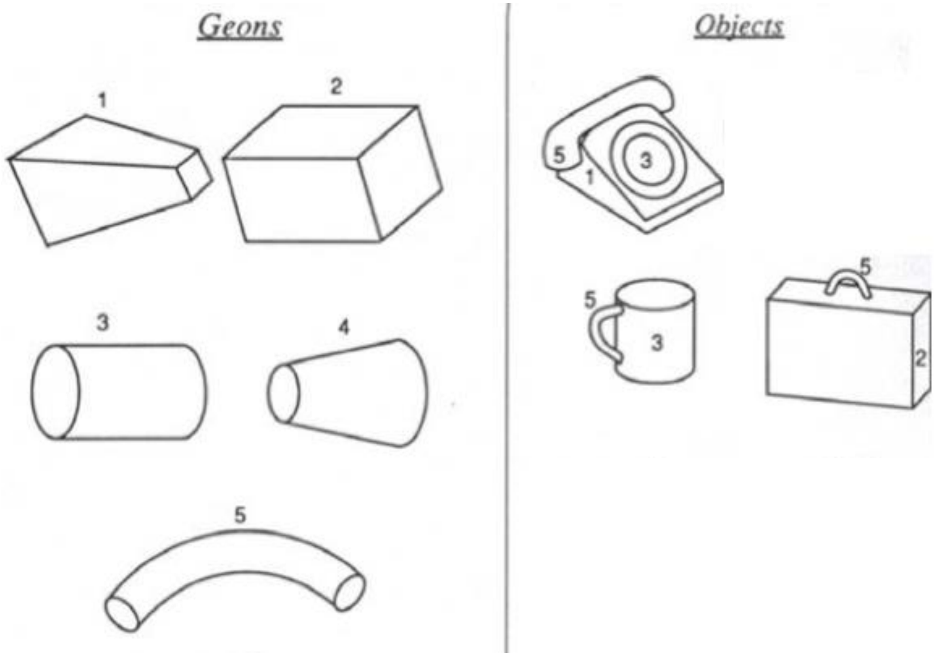
GEONS AND BIGRAM DETECTORS

- GEONS (geometry icons) are simple shapes
- From the network theory known as **recognition by components (RPC model)** (Hummel & Biederman 1992)
- According to Biederman (1987, 1880), we need (at most) 36 different GEONS to describe every object.



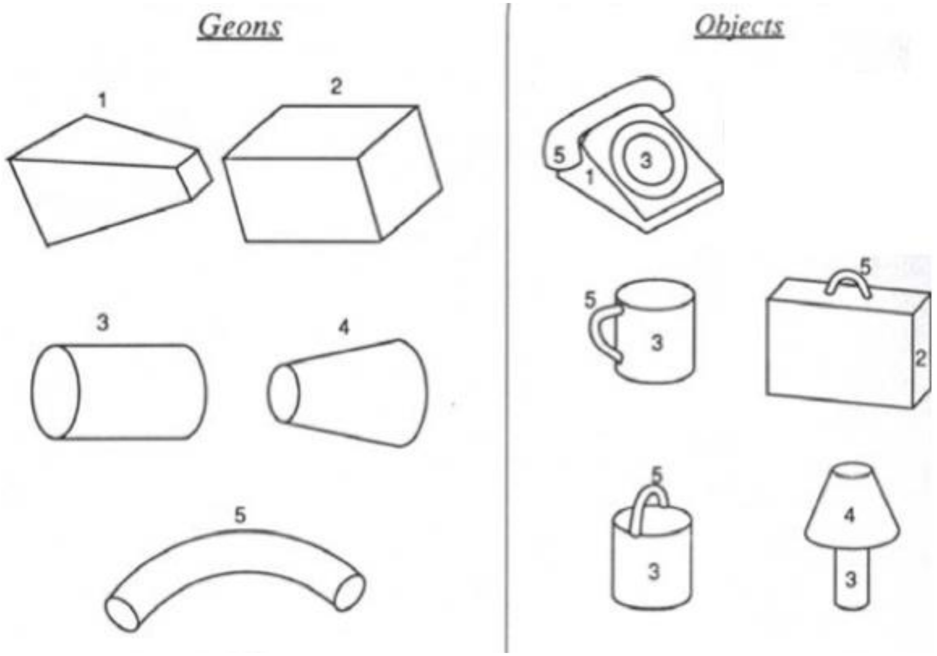
GEONS AND BIGRAM DETECTORS

- GEONS (geometry icons) are simple shapes
- From the network theory known as **recognition by components (RPC model)** (Hummel & Biederman 1992)
- According to Biederman (1987, 1880), we need (at most) 36 different GEONS to describe every object.



GEONS AND BIGRAM DETECTORS

- GEONS (geometry icons) are simple shapes
- From the network theory known as **recognition by components (RPC model)** (Hummel & Biederman 1992)
- According to Biederman (1987, 1880), we need (at most) 36 different GEONS to describe every object.



GEONS AND **BIGRAM DETECTORS**

- Let’s do a little experiment:
  Try to read the following letter combination:

GEONS AND
BIGRAM DETECTORS

- Let’s do a little experiment:
  Try to read the following letter combination:

“PIRT”

GEONS AND **BIGRAM DETECTORS**

- Let’s do a little experiment:

  Try to read the following letter combination:

GEONS AND **BIGRAM DETECTORS**

- Let's do a little experiment:
  Try to read the following letter combination:

“ITPR”

GEONS AND **BIGRAM DETECTORS**

- Let’s do a little experiment:
  Try to read the following letter combination:

GEONS AND
BIGRAM DETECTORS

- Let’s do a little experiment:
  Try to read the following letter combination:

GEONS AND **BIGRAM DETECTORS**

- Let’s do a little experiment:
  Try to read the following letter combination:

“HICE”

GEONS AND **BIGRAM DETECTORS**

- Let’s do a little experiment:
  Try to read the following letter combination:

GEONS AND **BIGRAM DETECTORS**

- Let’s do a little experiment:
  Try to read the following letter combination:

“HCEI”

GEONS AND **BIGRAM DETECTORS**

- Let’s do a little experiment:
  Try to read the following letter combination:

GEONS AND **BIGRAM DETECTORS**

- So why are “PIRT” or “HICE” easier to detect then “ITRP” and “HCEI”?
- We do not just detect words, but rather **bigrams**
- The **bigram detector** detects letter combinations and will be triggered by lower-level detectors

GEONS AND
BIGRAM DETECTORS

- So why are “PIRT” or “HICE” easier to detect then “ITRP” and “HCEI”?
- We do not just detect words, but rather **bigrams**
- The **bigram detector** detects letter combinations and will be triggered by lower-level detectors

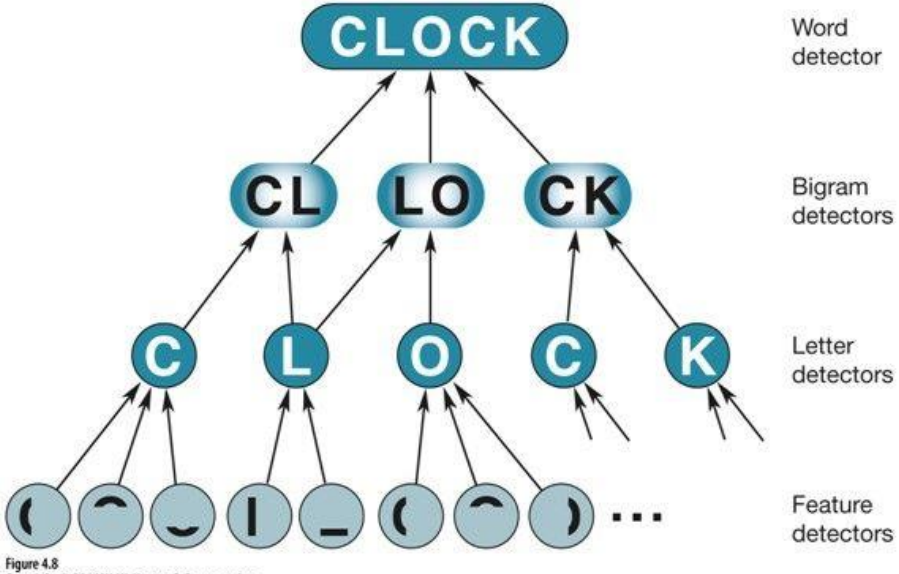


Figure 4.8
Copyright © 2016 W. W. Norton & Company, Inc.

GEONS AND **BIGRAM DETECTORS**

- So why are “PIRT” or “HICE” easier to detect then “ITRP” and “HCEI”?
- We do not just detect words, but rather **bigrams**
- The **bigram detector** detects letter combinations and will be triggered by lower-level detectors
- You have seen the letter pairs “HI” (“HIT”, “HIGHT”) and “CE” (“FACE”, “MICE”) before
  - **Benefits from priming**

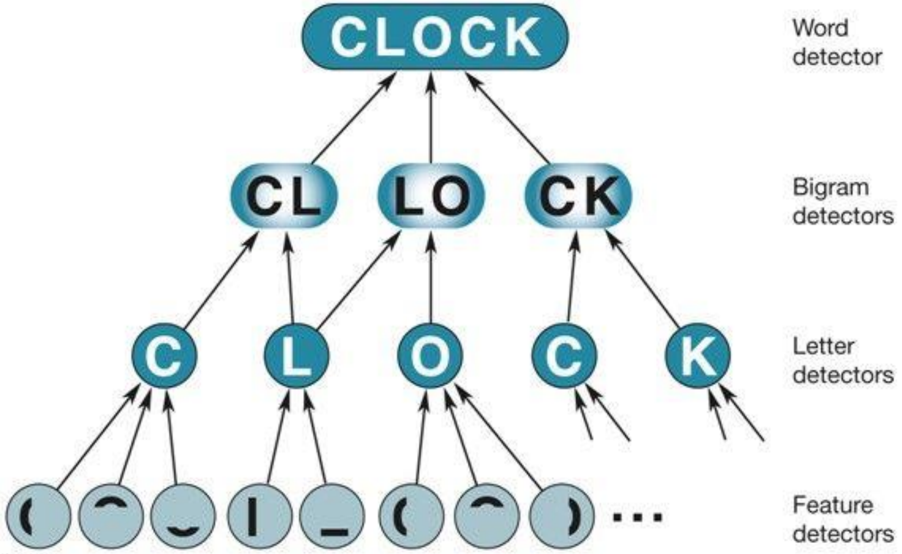
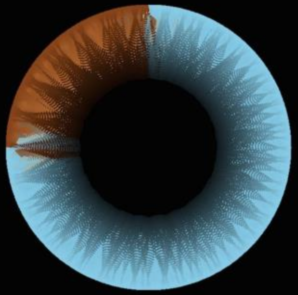


Figure 4.8
Copyright © 2016 W. W. Norton & Company, Inc.

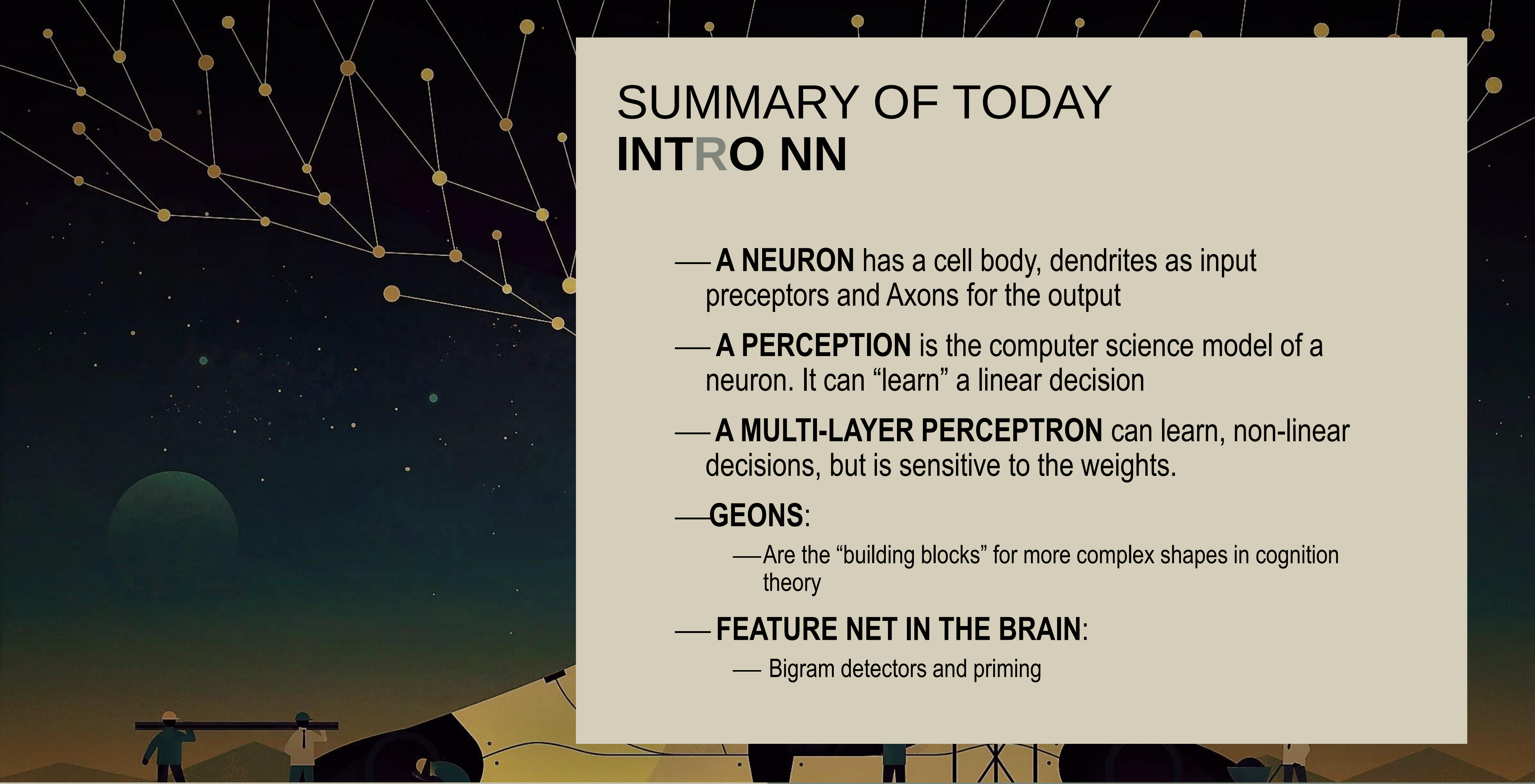
FEATURE **DETECTION IN IMAGES**

FEATURE **DETECTION IN IMAGES**



https://www.youtube.com/watch?v=aircAruvnKk

until ~13:15



SUMMARY OF TODAY
INTRO NN

- **A NEURON** has a cell body, dendrites as input preceptors and Axons for the output
- **A PERCEPTION** is the computer science model of a neuron. It can “learn” a linear decision
- **A MULTI-LAYER PERCEPTRON** can learn, non-linear decisions, but is sensitive to the weights.
- **GEONS**:
  - Are the “building blocks” for more complex shapes in cognition theory
- **FEATURE NET IN THE BRAIN**:
  - Bigram detectors and priming