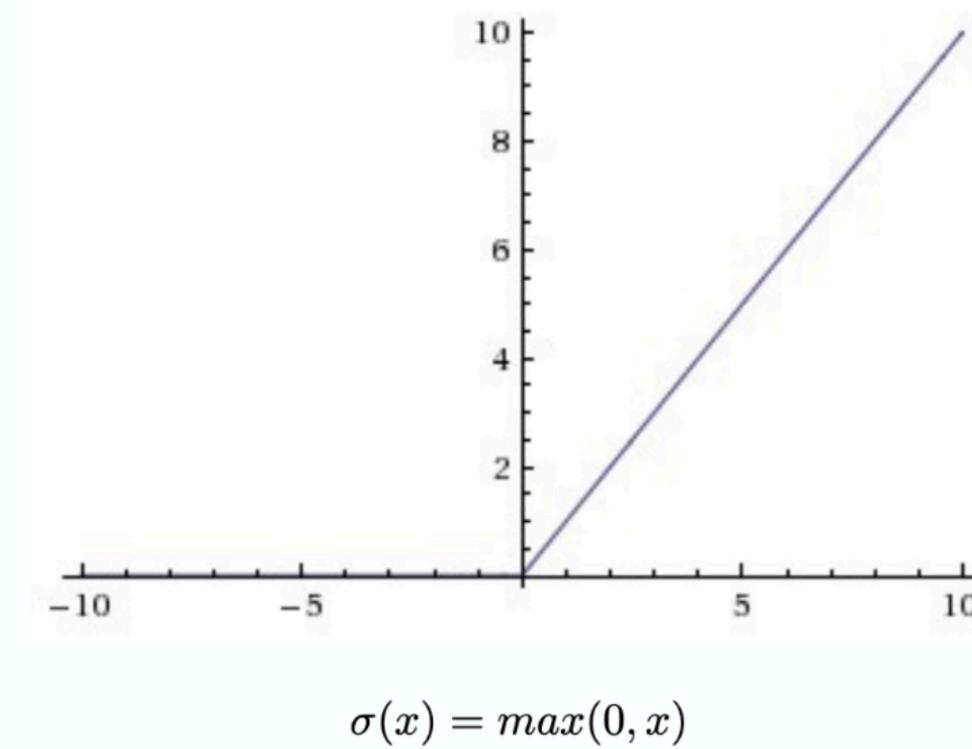
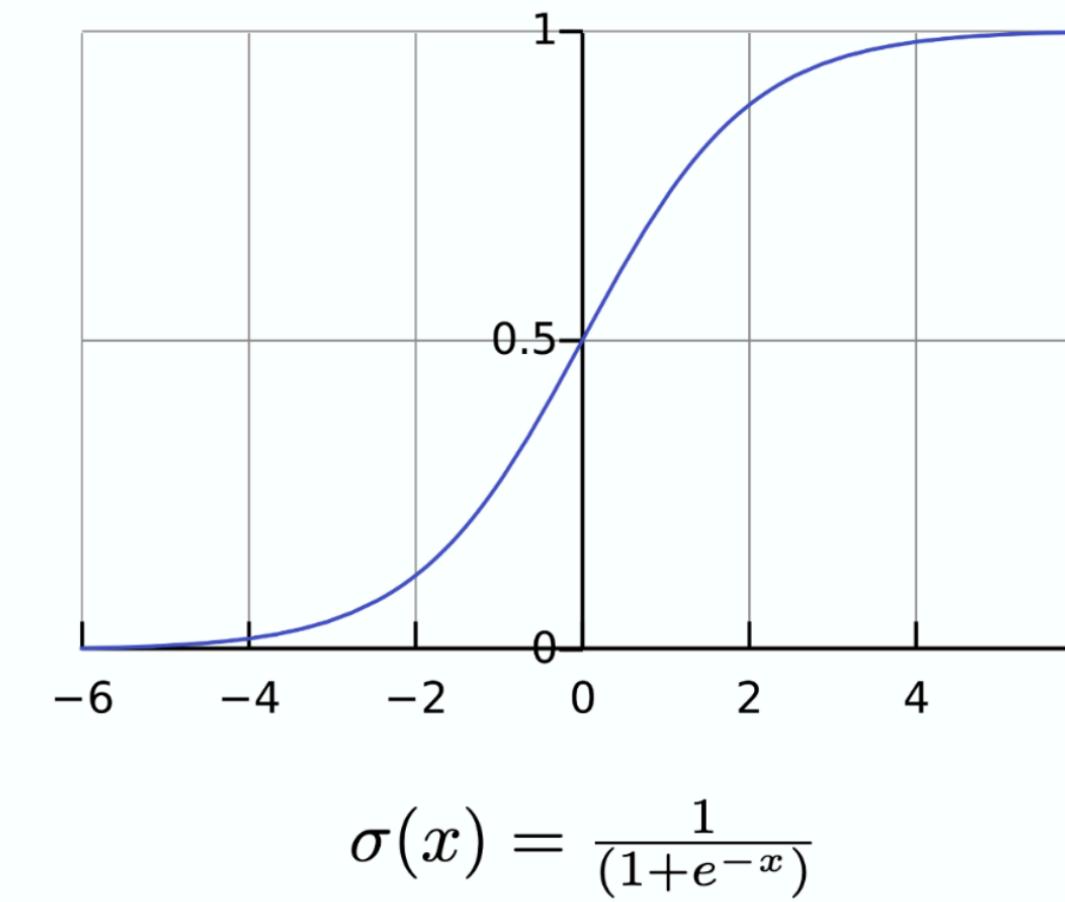
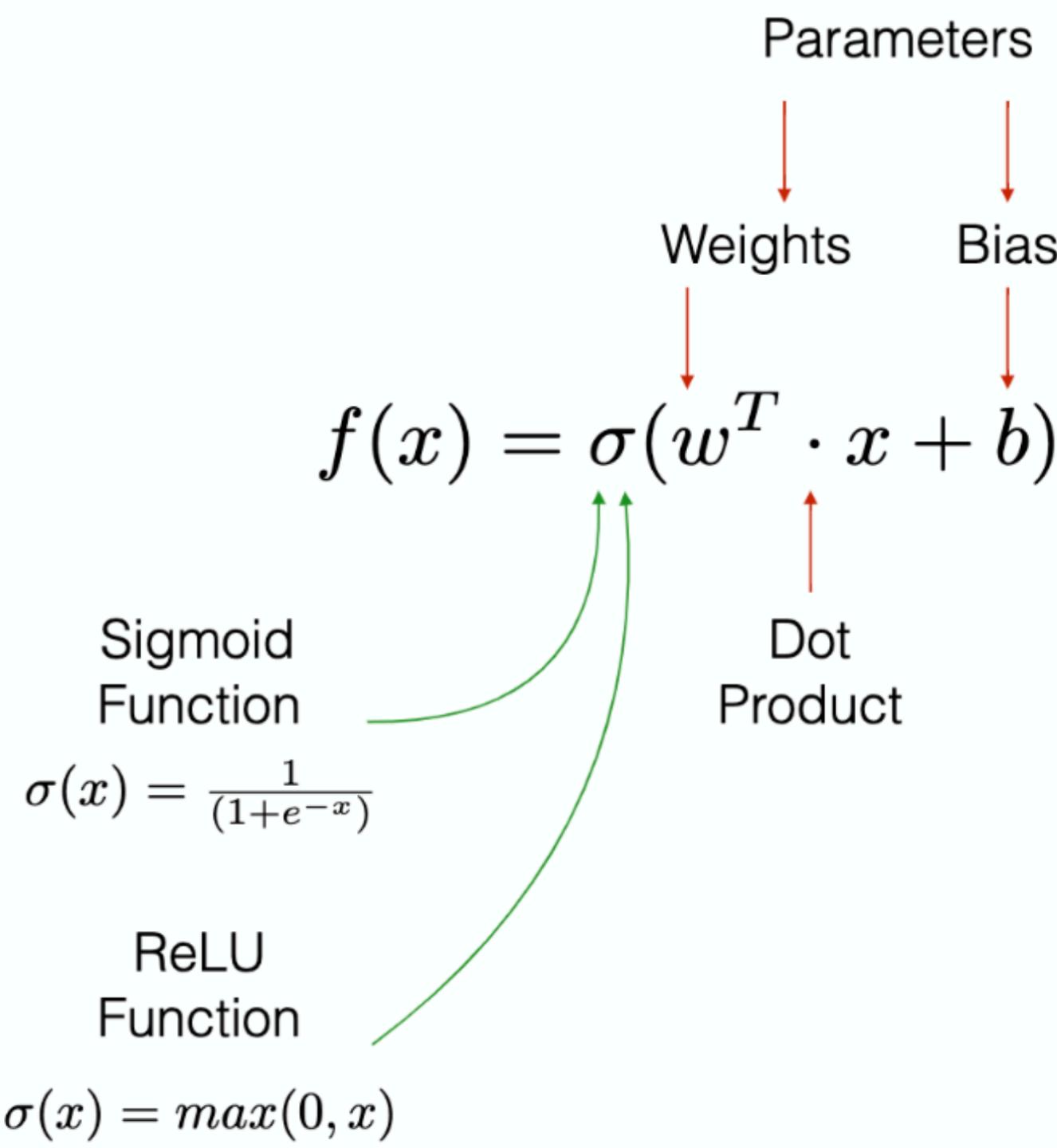


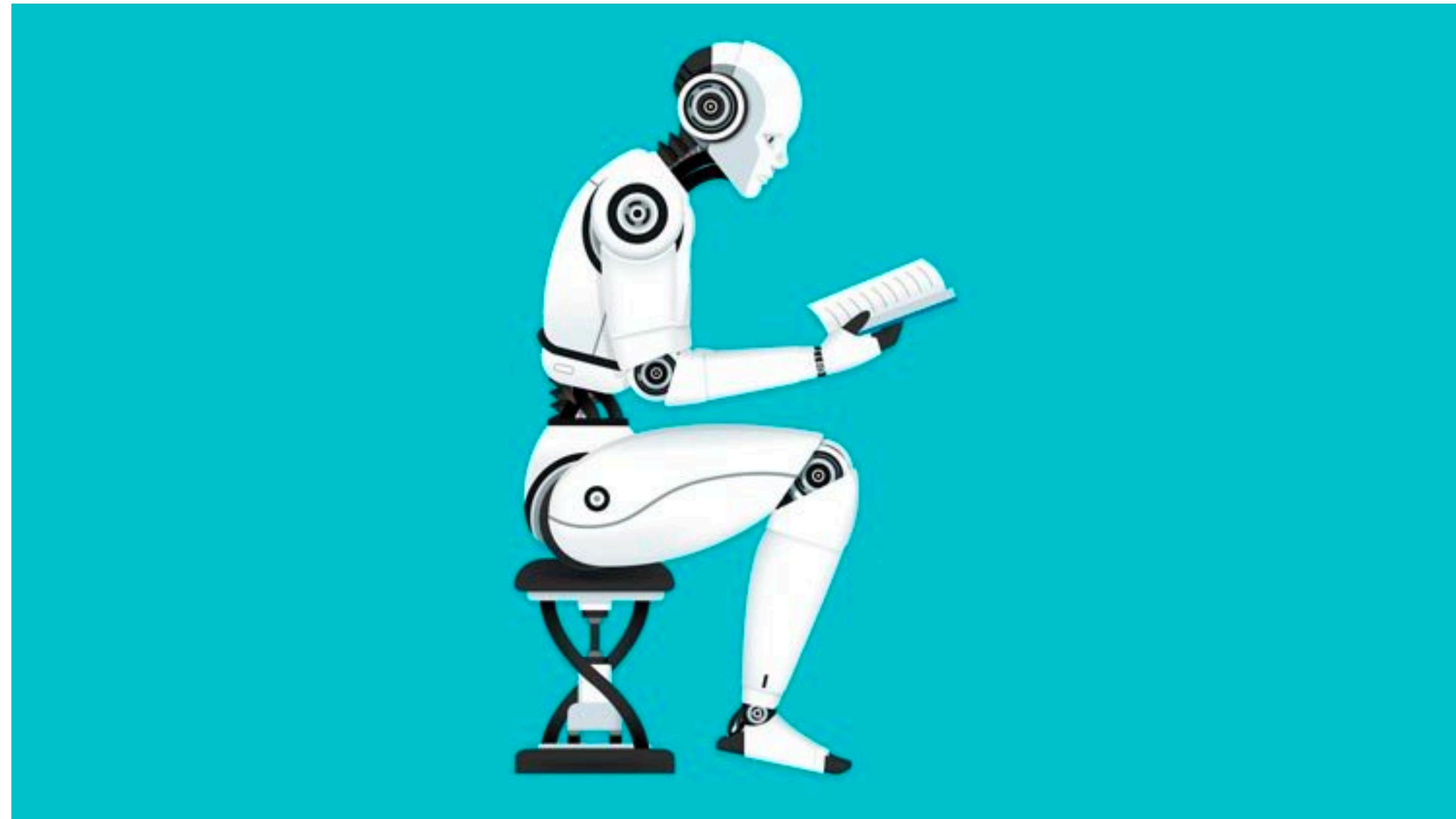
# Neural Networks

## 1 Layer Neural Net model





UNIVERSITAT DE  
BARCELONA



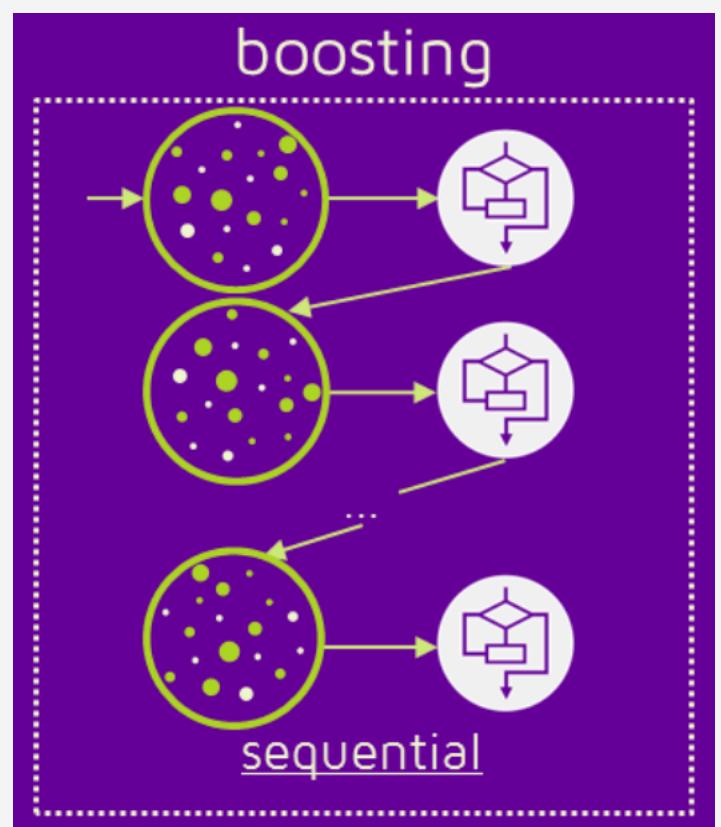
# Neural Networks

Machine Learning | Enginyeria Informàtica

Santi Seguí | 2022-2023

## **Project 4: Build a boosting classifier from scratch**

In this project you have to built a Boosting Classifier Model from scrath. `DecisionTreeClassifier` from `sklearn` can be used as weak model. (You don't need to implement it)

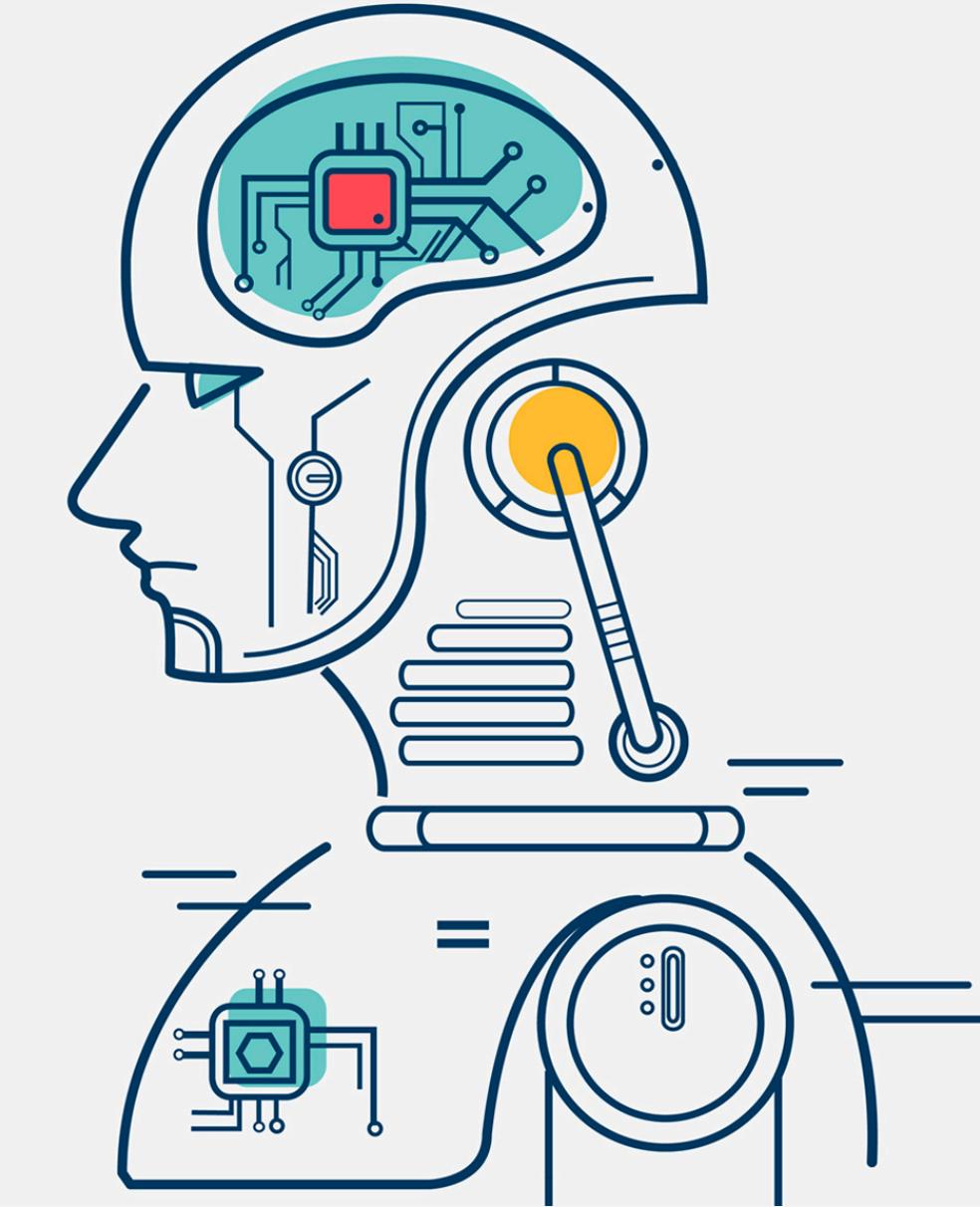


### **Requirements:**

- 1) Evaluate your model using MNIST & Fashion MNIST DATASET.
- 2) Compare the results with Bagging; Random Forest; AdaBoost (you can use `sklearn` models)

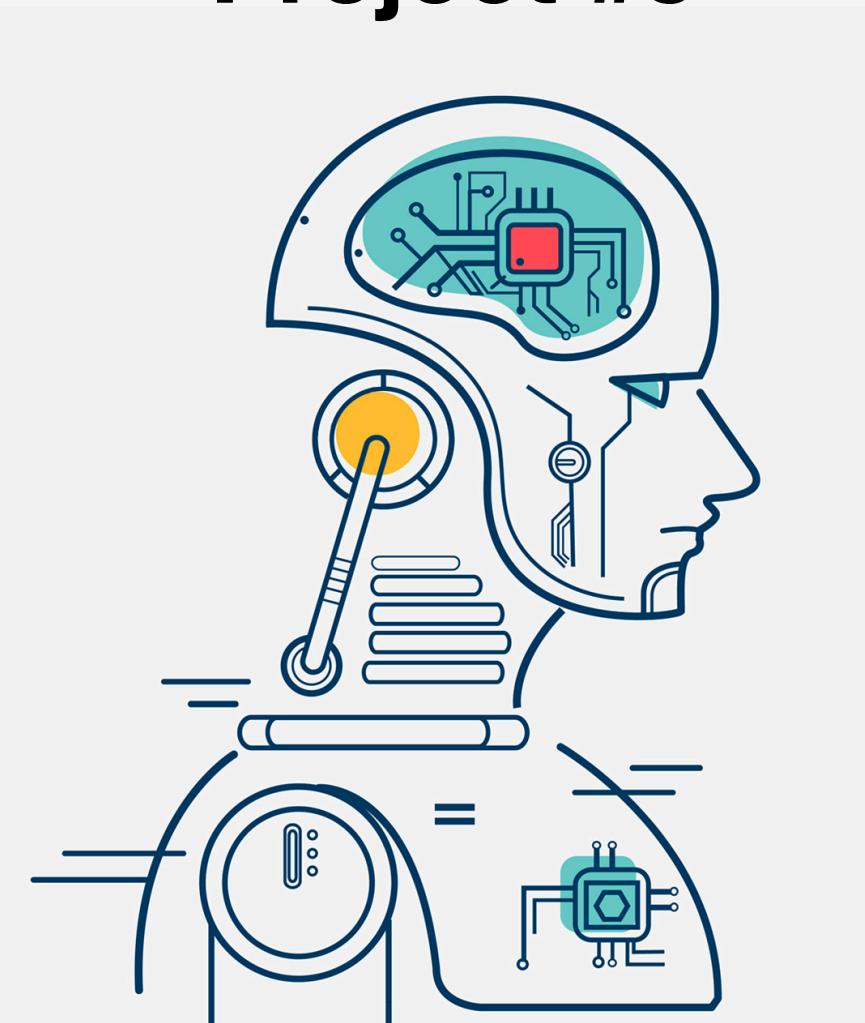
### **Score:**

4 points

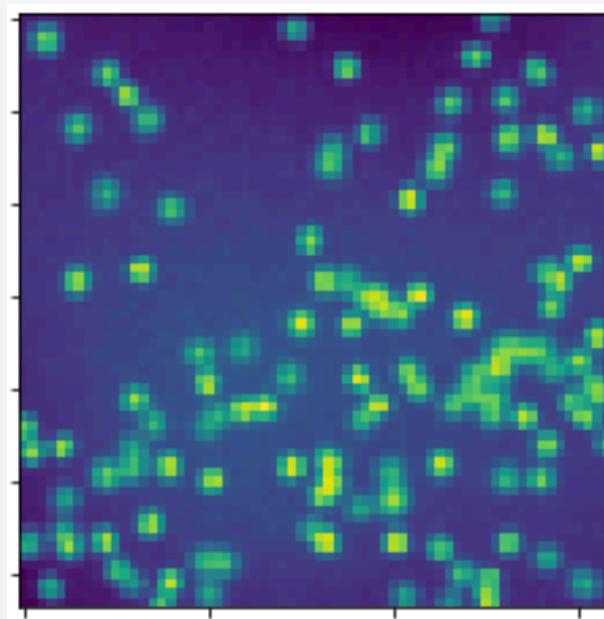


**Project #4**

## Project #5



In this image appears 126 dots



Neural Networks for **Learning to Count Problem**

The **output** is the **number of instance of an object** that appears in that image ->

**Score:**

What is expected:

- \* Good Result
- \* CNN architecture
- \* Some kind of Data Augmentation:

Score: 4 points

<https://www.kaggle.com/t/208b522b812d4de3b7db184a1ac194d9>

# What is a Neural Network

- It is a system **biologically inspired** that tries to emulate **human brain**.
- **Why**

# What is a Neural Network

- It is a system **biologically inspired** that tries to emulate **human brain**.
- **Why** is it a good idea to try to **emulate** the **brain** when solving a recognition task?

# What is a Neural Network

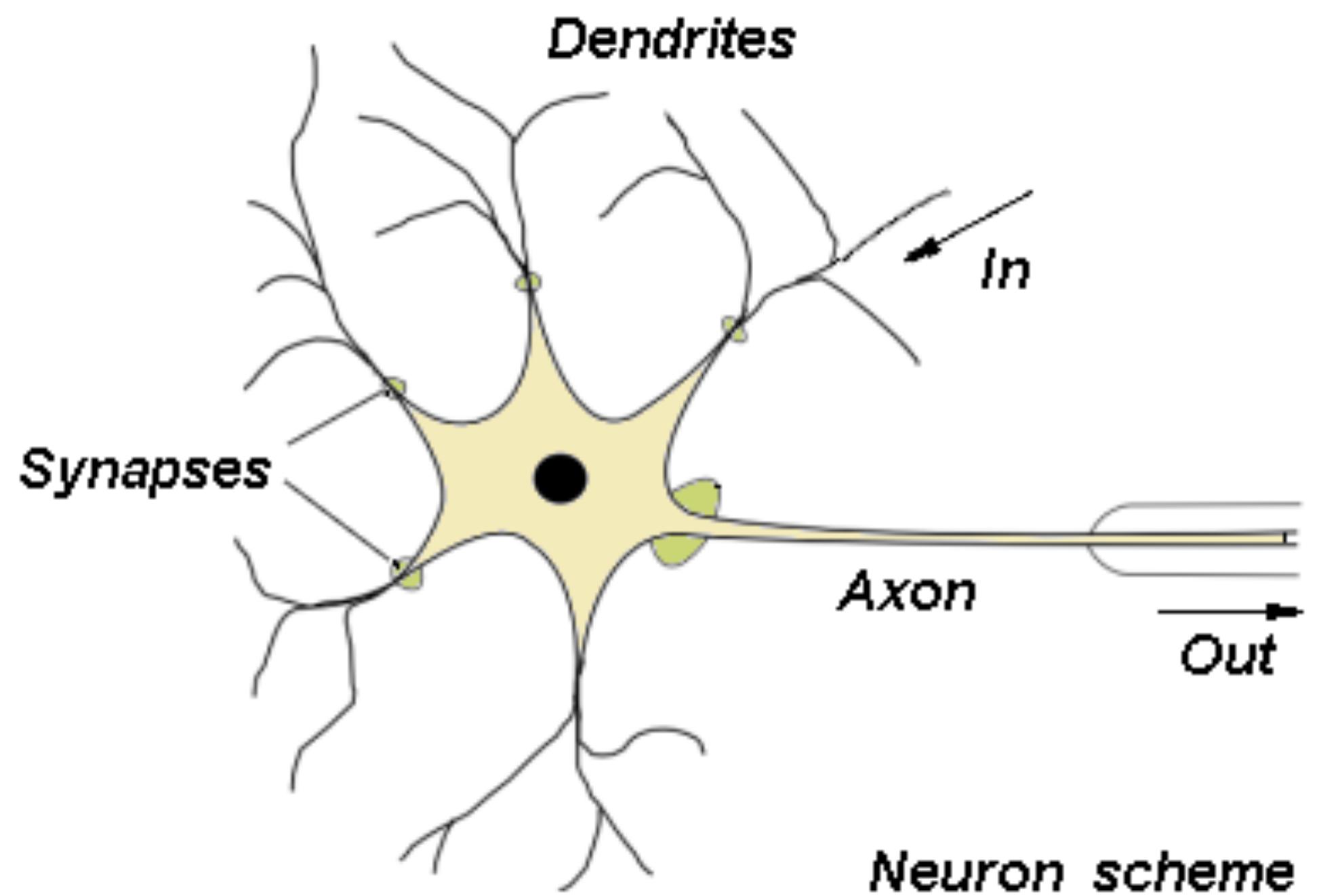
- It is a system **biologically inspired** that tries to emulate **human brain**.
- **Why** is it a good idea to try to **emulate** the **brain** when solving a recognition task?

The human brain is the **best system** that we know of!!

# What is a neuron?

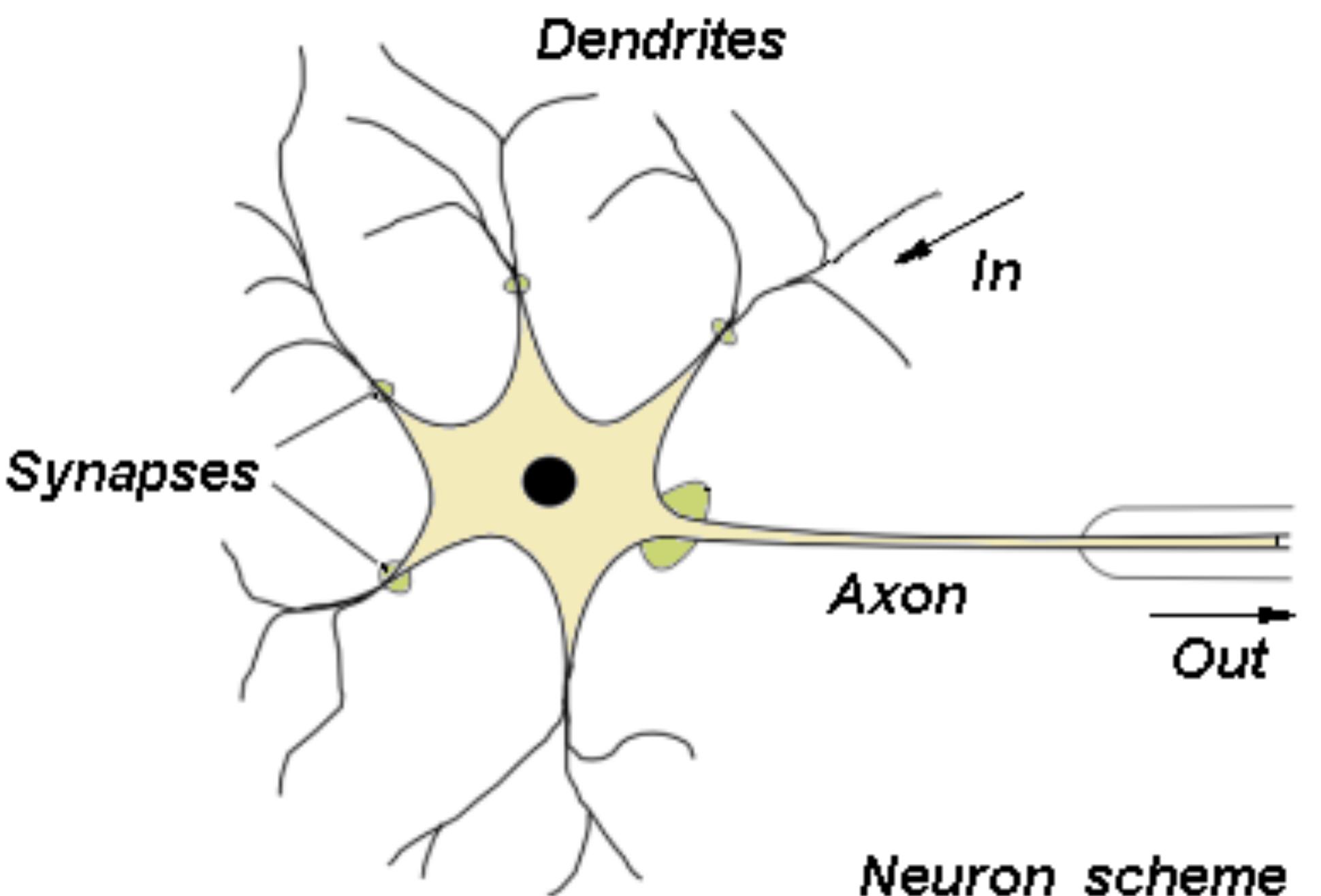


# What is a neuron?



# What is a neuron?

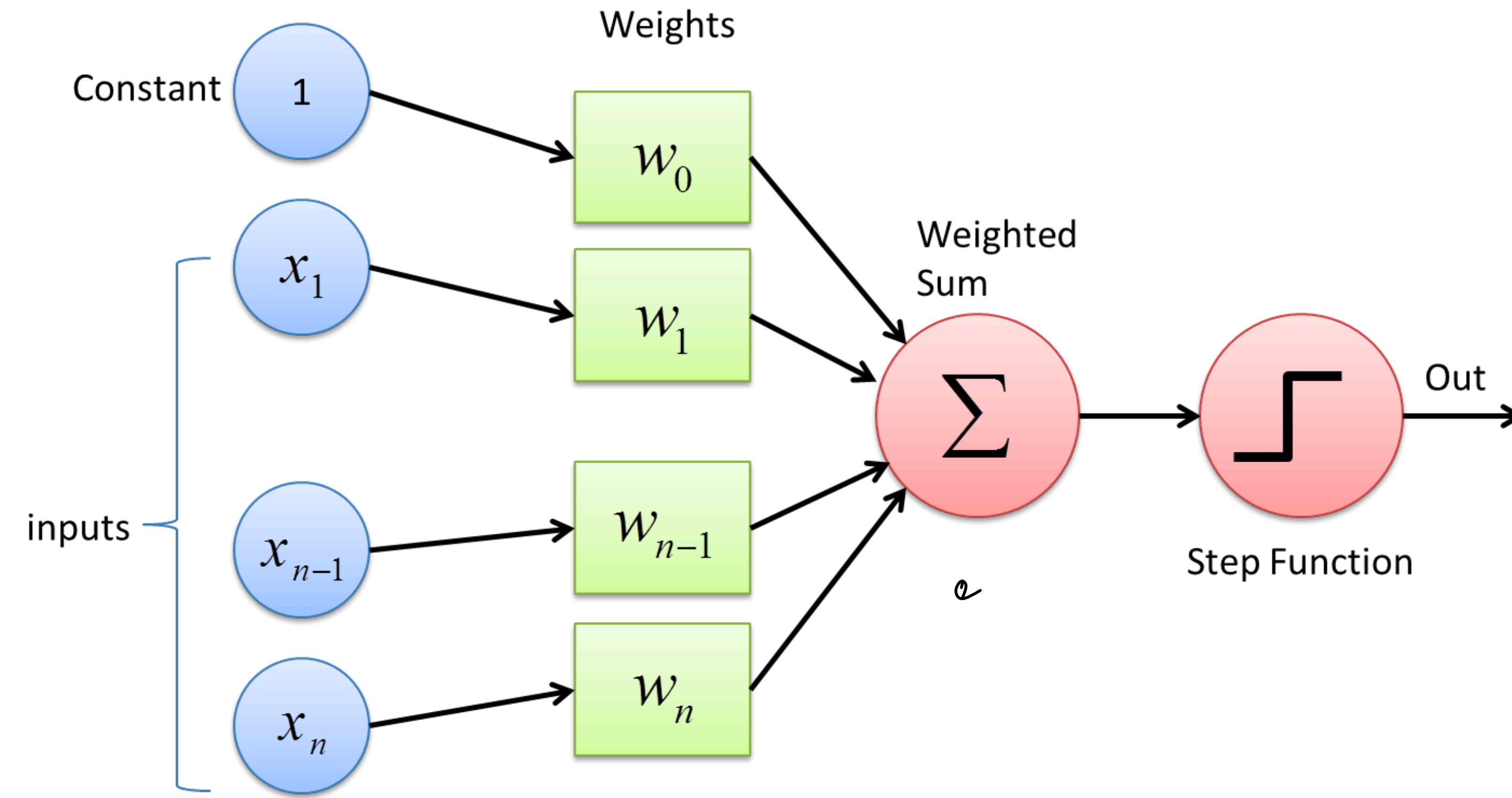
- In the human brain, a **typical neuron** collects signals from others through a host of fine structures called **dendrites**.
- The neuron **sends out spikes** of electrical activity through a long, thin stand known as an **axon**, which splits into thousands of branches.
- At the end of each branch, a structure called a **synapse** converts the activity from the axon into electrical effects that inhibit or excite activity in the connected neurons.



*Neuron scheme*

# How the brain works (with one slide!)

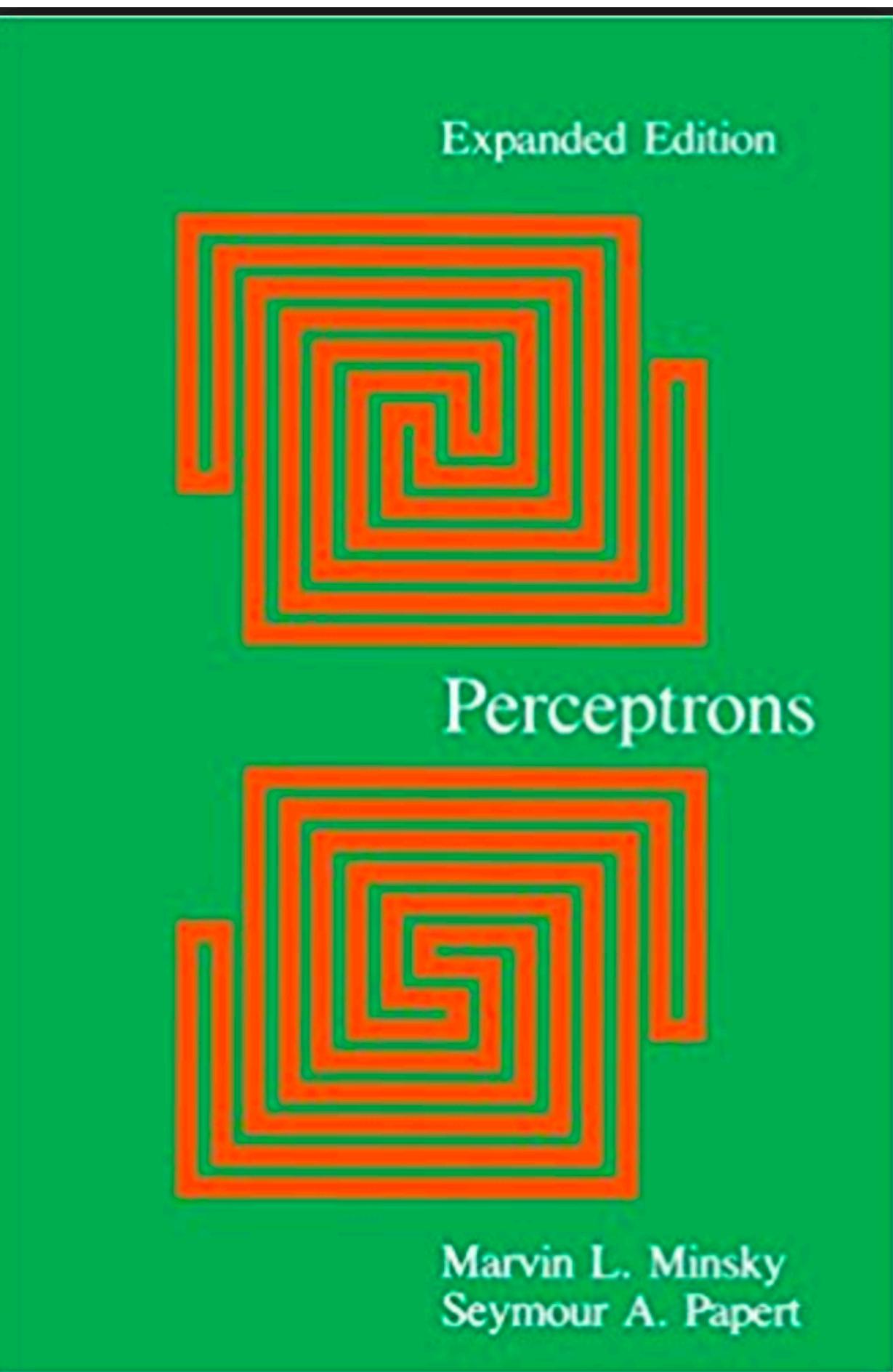
- Each neuron receives inputs from other neurons
  - A few neurons also connects to receptors
  - Cortical neurons use spikes to communicate
- The effect of each input line on the neuron is controlled by a synaptic weight
  - The weights can be positive or negative
- The synaptic weight adapt so that the whole network learns to perform useful computations
  - Recognizing objects, understanding language, making plans, controlling the body.
- You have about  $10^{11}$  neurons each with about  $10^4$  weights.
  - A huge number of weights can affect the computation in a very short time. Much better bandwidth than a workstation.

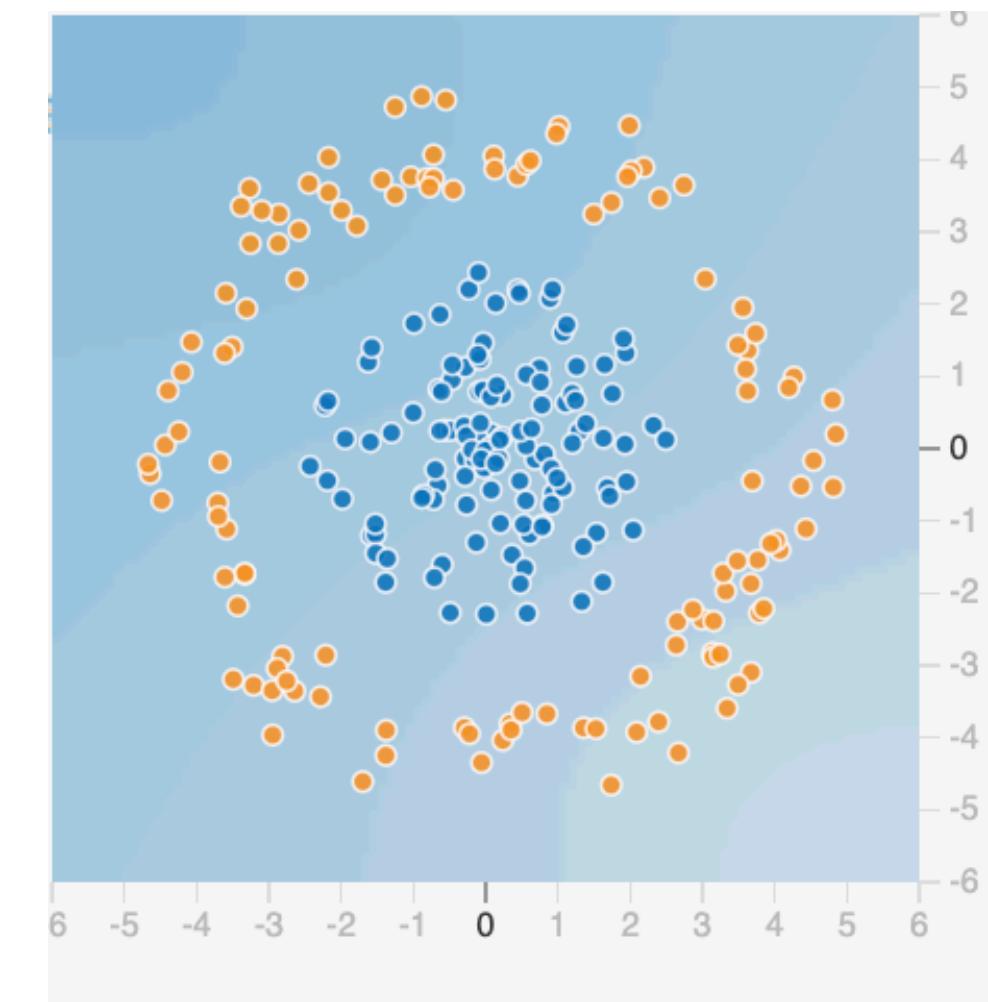
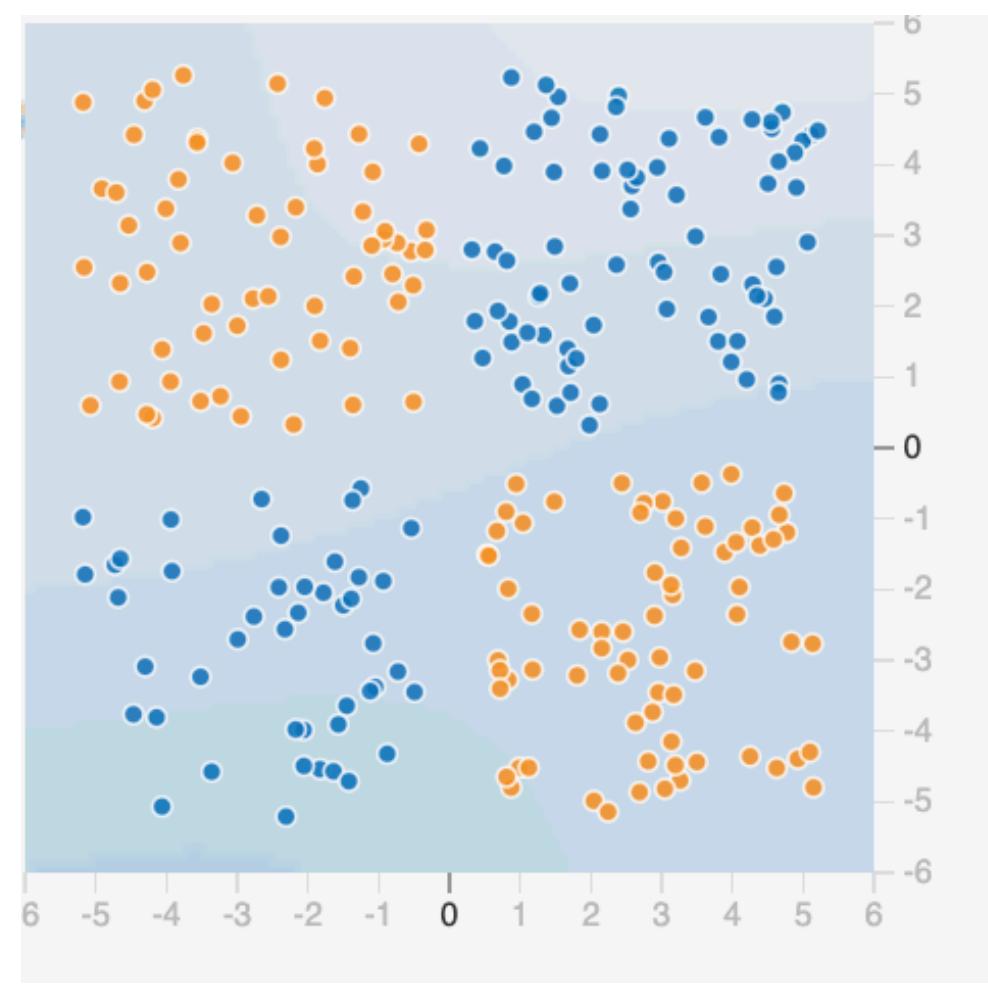
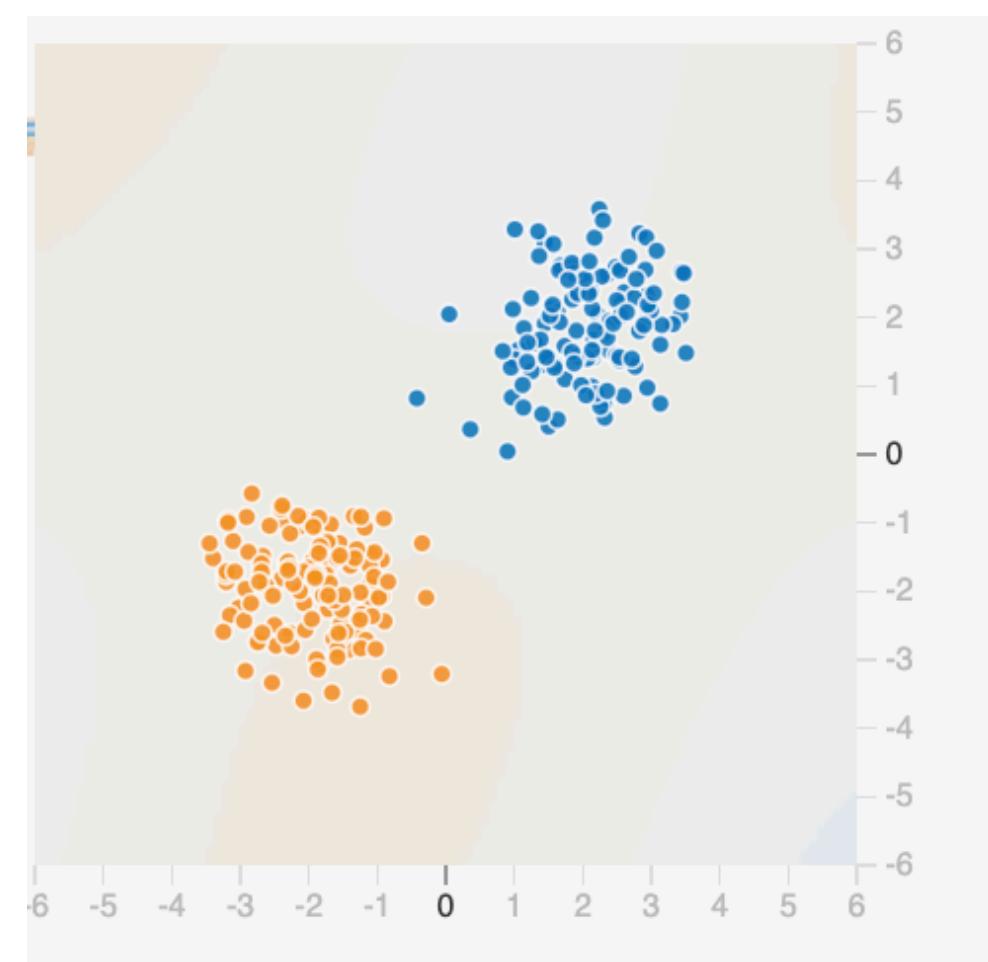


# Perceptron

Perhaps the first AI model

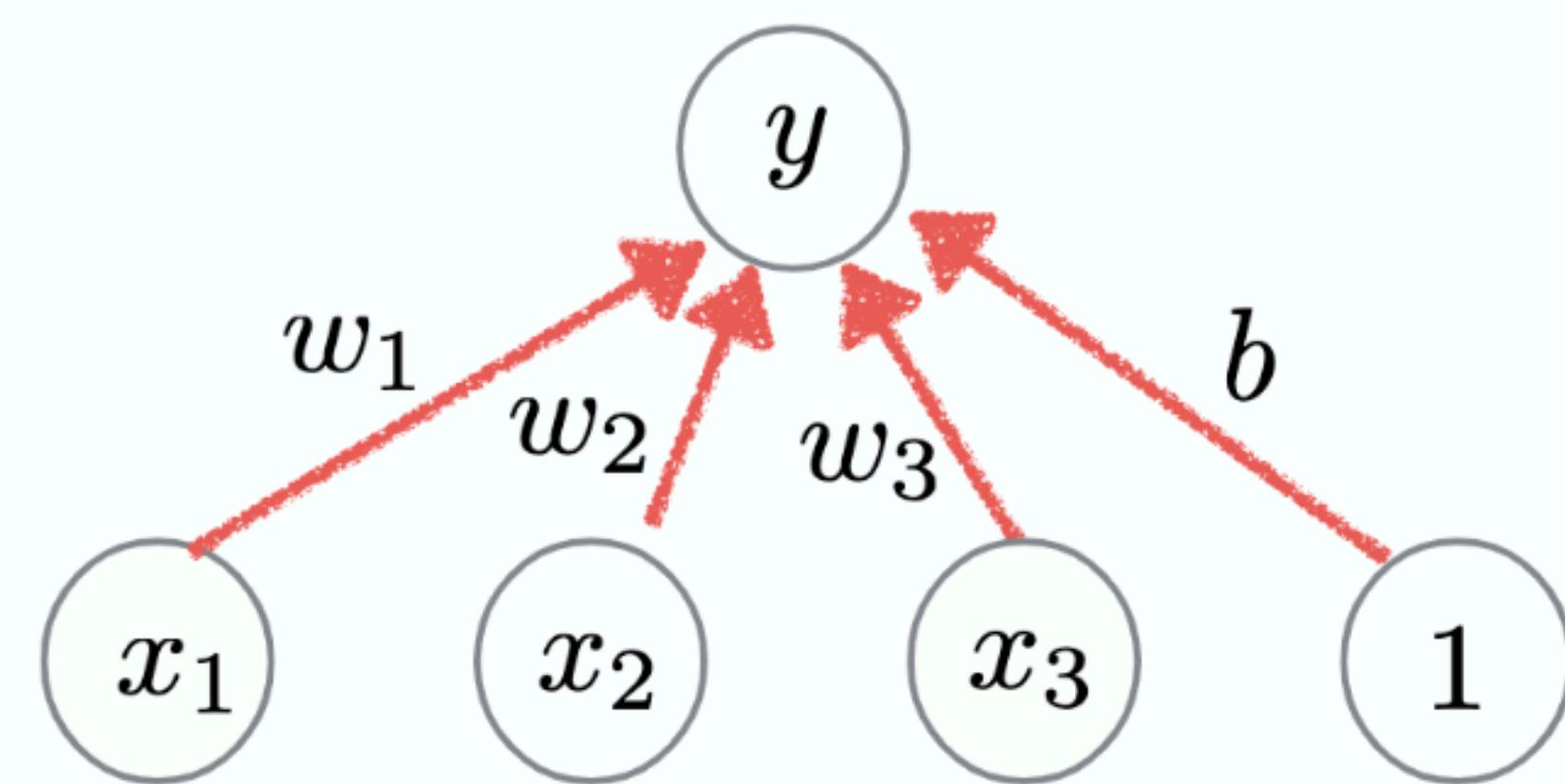
# The Book: Perceptrons





# 1 Layer Neural Net model

$$f(x) = \sigma(w^T \cdot x + b)$$



Graphical Representation

# Different types of Neurons

- Linear Neurons
  - These are one of the most simple neurons models computationally limited

$$y = \beta + \sum_i X - i\beta_i$$

# Different types of Neurons

- McCulloch-Pitts (Binary threshold neurons):
  - First compute a weighted sum of the inputs
  - Then send out a fixed size spike of activity if the weighted sum exceeds a threshold.

$$y = \beta_0 + \sum_i x_i \beta_i$$

$$y = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

# Different types of Neurons

- Rectified Linear Neurons (ReLU)
  - It computes a linear weights sum of their inputs.
  - The output is a non-linear function of the total input.

$$y = \beta_0 + \sum_i x_i \beta_i$$

$$y = \begin{cases} z, & \text{if } z \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

# Different types of Neurons

- Sigmoid neurons
- These give a real-valued output that is a smooth and bounded function of the total input. Logistic function is typically used.

$$y = \beta_0 + \sum_i x_i \beta_i$$

$$y = \frac{1}{1 + e^{-z}}$$

# Perceptron Multiclasse

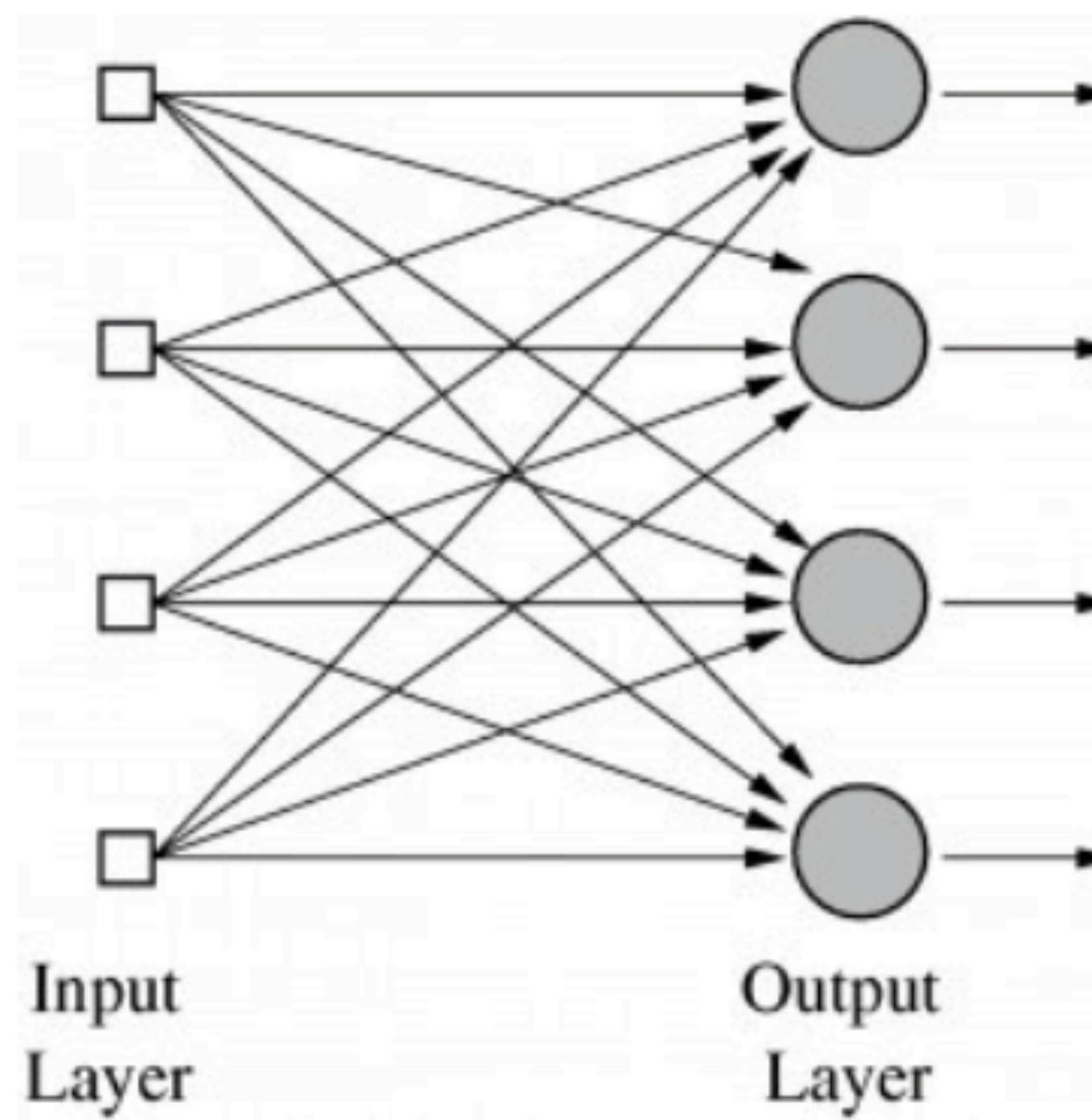


Figura 4: Modelo del perceptor para clasificación múltiple

# Exercice

- Suppose we have a 3-dimensional input  $x = (x_1, x_2, x_3)$  connected to a neuron with weights  $w = (w_1, w_2, w_3)$  where :

$$x_1 = 2 \quad w_1 = 1$$

$$x_2 = -1 \quad w_2 = -0.5$$

$$x_3 = 1 \quad w_3 = 0$$

and a bias  $b = 0.5$ .

- For each of the types of neurons discussed so far, we calculate the output  $y$  using the input  $x$ , weight  $w$  and bias  $b$ . If the activation function is:

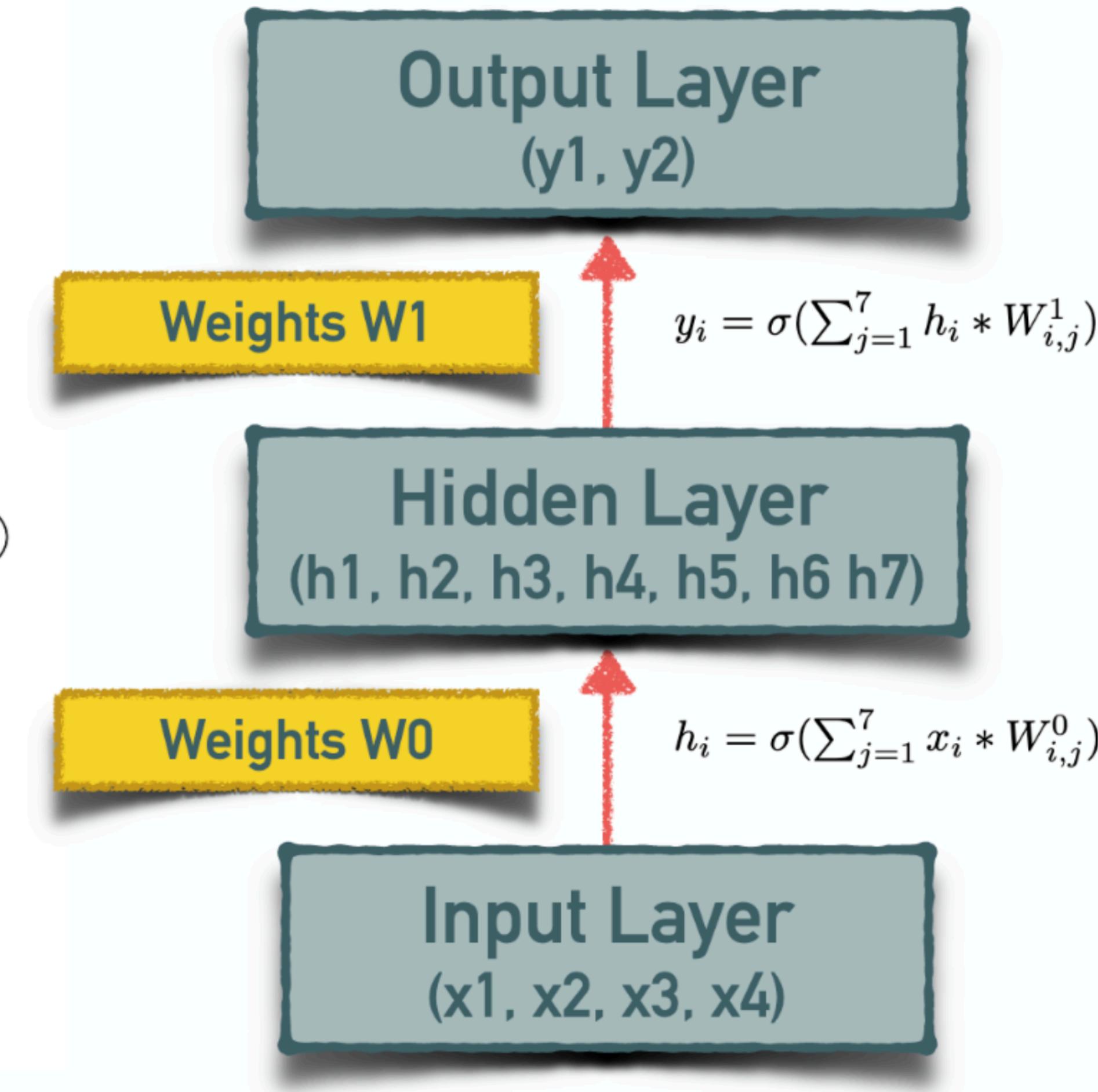
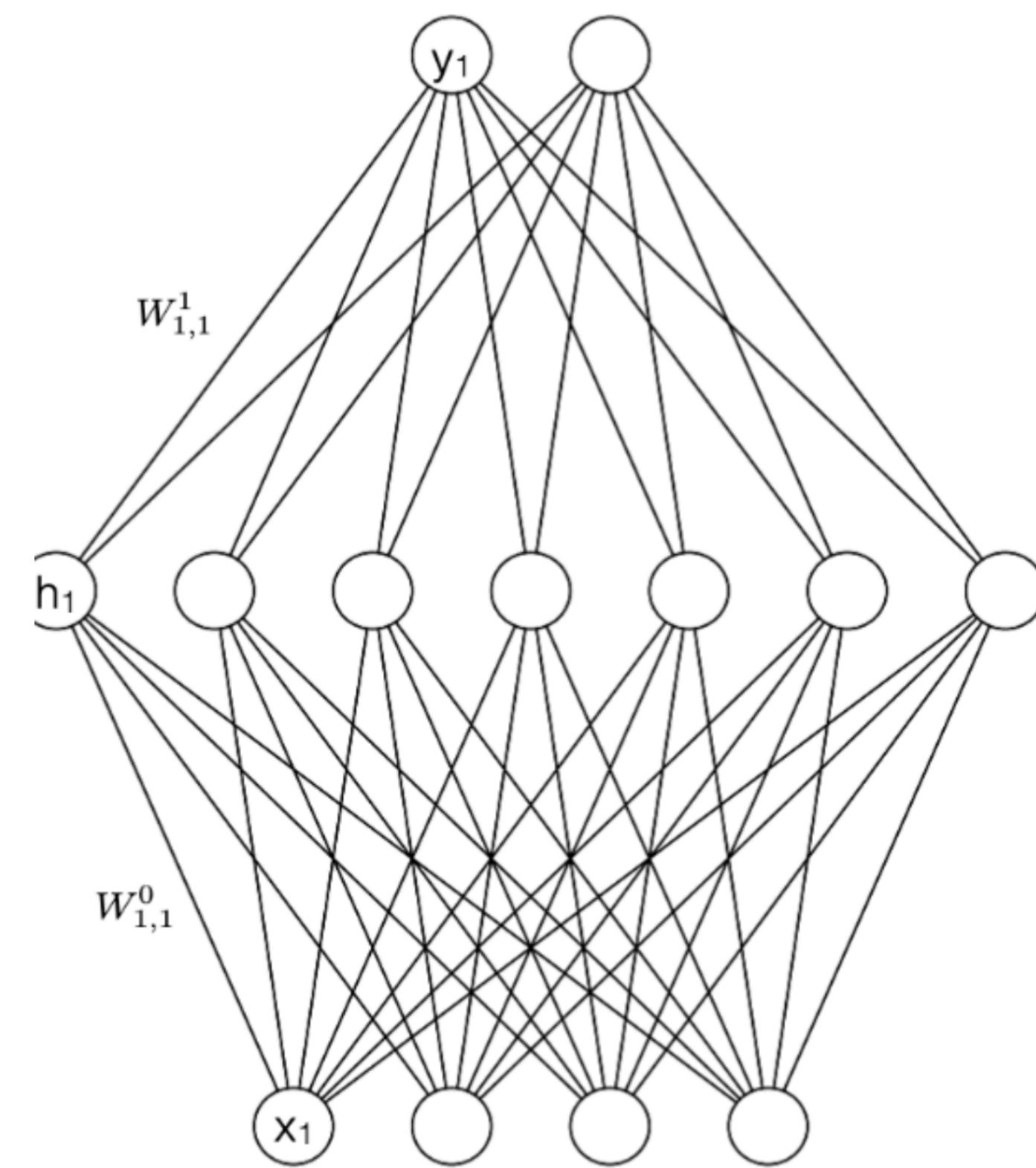
- Linear
- Logistic Sigmoid
- Binary Threshold
- Rectified Linear

# Types of Neural Networks

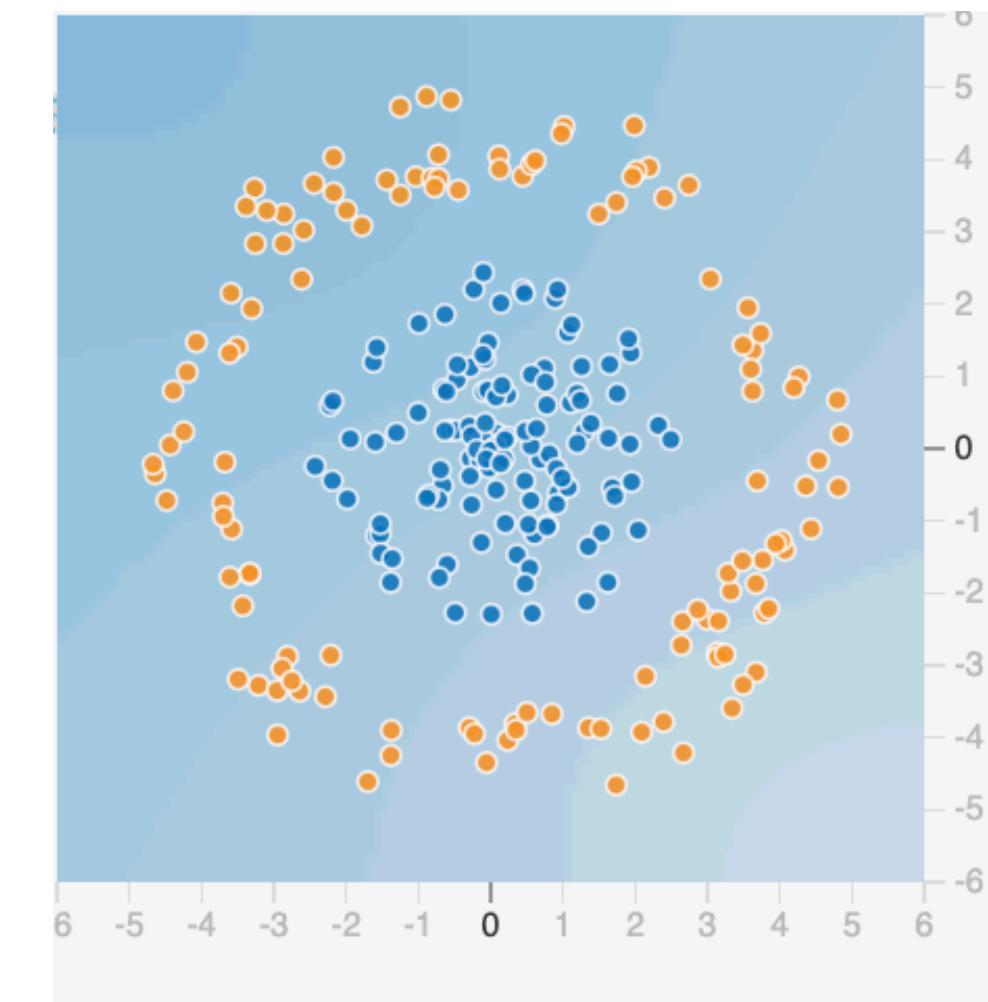
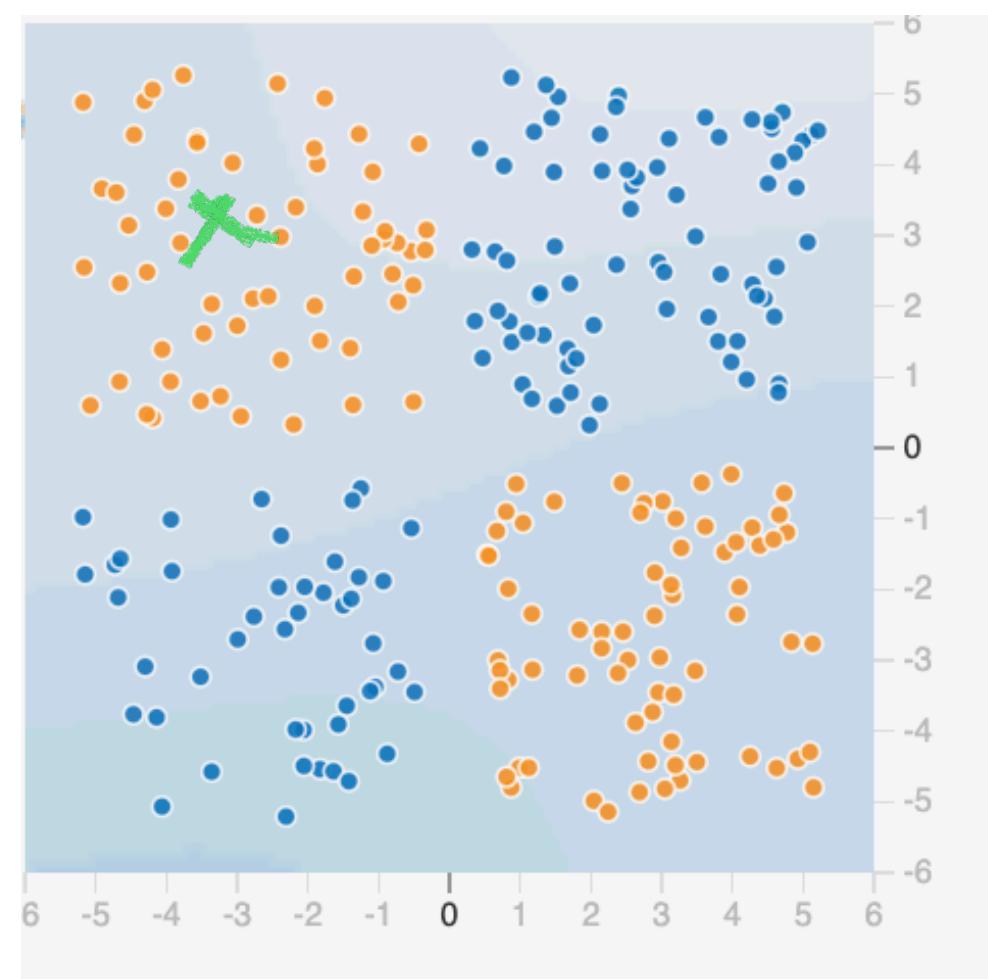
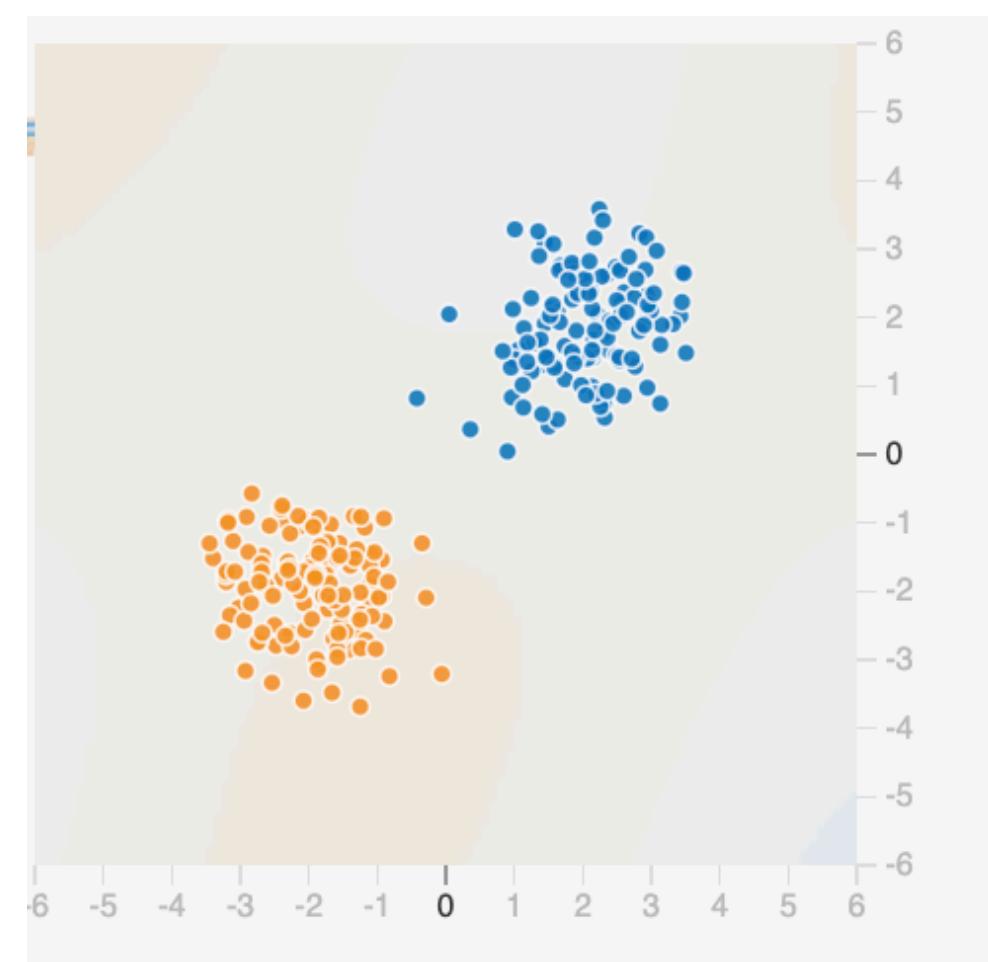
- Feed Forward Neural Networks
- Convolutional Neural Networks
- Recurrent Neural Networks

# Feed-Forward Neural Network

- These are the commonest type of neural networks in practical applications
  - The first layer is the input and the last layer is the output
  - If there is more than one hidden layer, we call them "deep" neural networks!
- They compute a series of transformations that change the similarities between cases.
  - The activities of the neurons in each layer are non-linear functions of the activities in the layer below.



## 2 Layers Neural Net model



# Let's play a little bit! :)

- <http://playground.tensorflow.org/>