# Advanced School in Artificial Intelligence

# Introduction to Neural Networks

Ing. Zese Riccardo
riccardo.zese@unife.it

Regione Emilia-Romagna

*Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021*

Università degli Studi di Ferrara

# Outline

- Introduction to Python
- Introduction to Neural Networks
- Convolutional NN
- Recurrent NN
- Autoencoders and self supervised learning

*Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021*

Regione Emilia-Romagna

Università degli Studi di Ferrara

# Sources

- These slides are taken from:
  - Intel Nervana AI Academy Intructional Content

    
  - Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

    https://www.deeplearningbook.org/lecture_slides.html
  - Chapter "Neural Networks" of "A Course in Machine Learning" by Hal Daume III.

    http://ciml.info/
  - Some parts from "CS231n: Convolutional Neural Networks for Visual Recognition", Stanford University

    http://cs231n.stanford.edu/

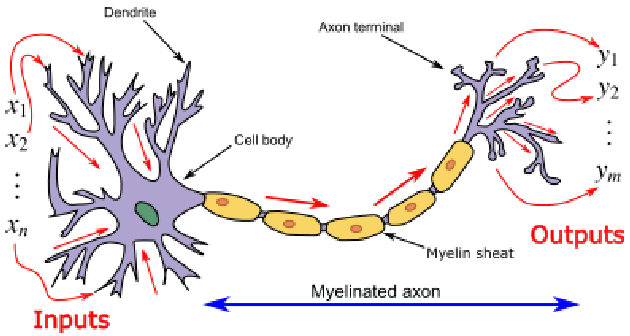
# Outline

- Introduction to Python
- Introduction to Neural Networks
- Convolutional NN
- Recurrent NN
- Autoencoders and self supervised learning

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

# Neural Networks

- The human brain is composed of about $10^{11}$ **neurons** (100 billion).
- Each neuron is connected to others and receives electrical signals through the **dendrites**.
- Each neuron sends messages through a long dendrite appendix called an **axon**.
- At the end of each axon a series of ramifications branch out called **synapses**.
- The synapses have the purpose of **stimulating** the other neurons, through electrical signals, in order to activate them.
- **Learning** influences the ability of synapses to activate other neurons or not.




Img from Wikipedia, CC BY-SA 4.0, Prof. Loc Vu-Quoc. https://en.wikipedia.org/wiki/Artificial_neural_network#/media/File:Neuron3.png

# Neural Networks

- Two different reasons have historically driven the study of **Artificial Neural Networks** (ANN):
  - Reproduction (and therefore understanding) of the human brain by building reliable models of all or at least part of the brain, reproducing neuro-physiological phenomena, and performing experimental verification (does the proposed model reproduce biological data?)
  - Extraction of the fundamental principles of calculation used by the human brain to produce artificial systems, possibly different from the brain, that reproduce its functions, perhaps in a faster and more efficient way.

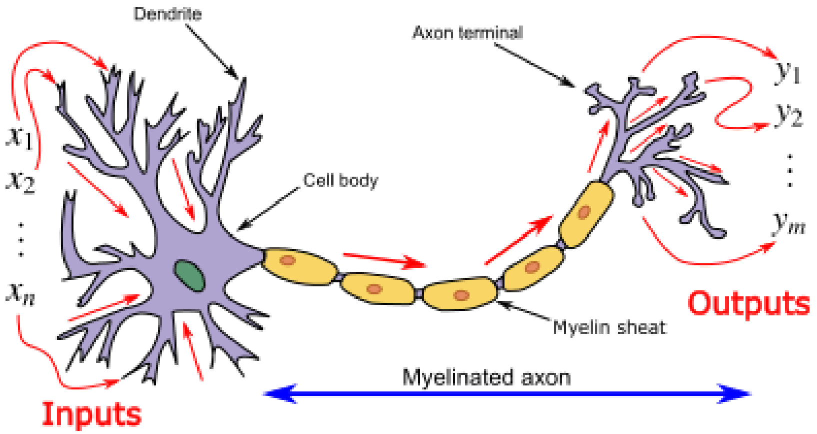
# A bit of history…

- 1943 McCulloch and Pitts: first mathematical model of an NN, binary input, linear threshold combination, binary output.
- 1958 first neural network scheme, called Perceptron, forerunner of the current neural networks. Great enthusiasm.
- 1969 Minsky and Papert show limits of Perceptron. Research in this field has a considerable drop in funding.
- 1986 Rumelhart and al. proposed the Backpropagation algorithm for learning the weights of a multilayer network. New life to research.

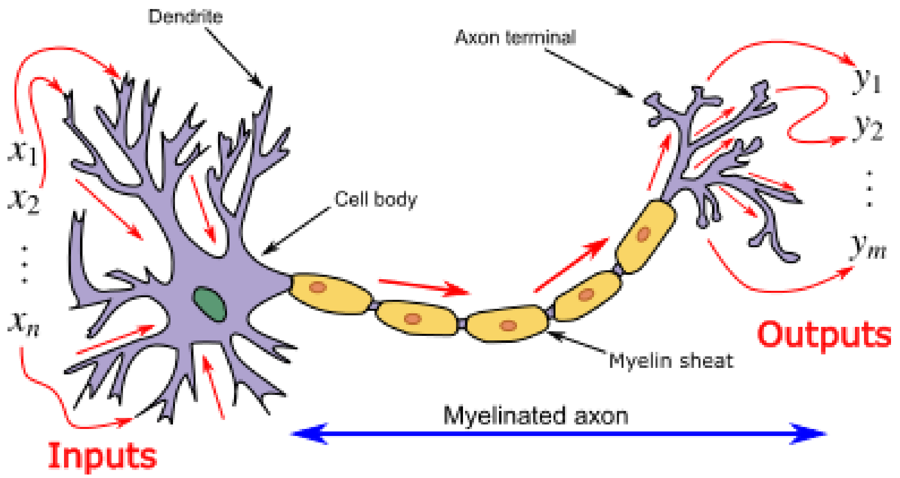

*Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021*

# Neural Networks

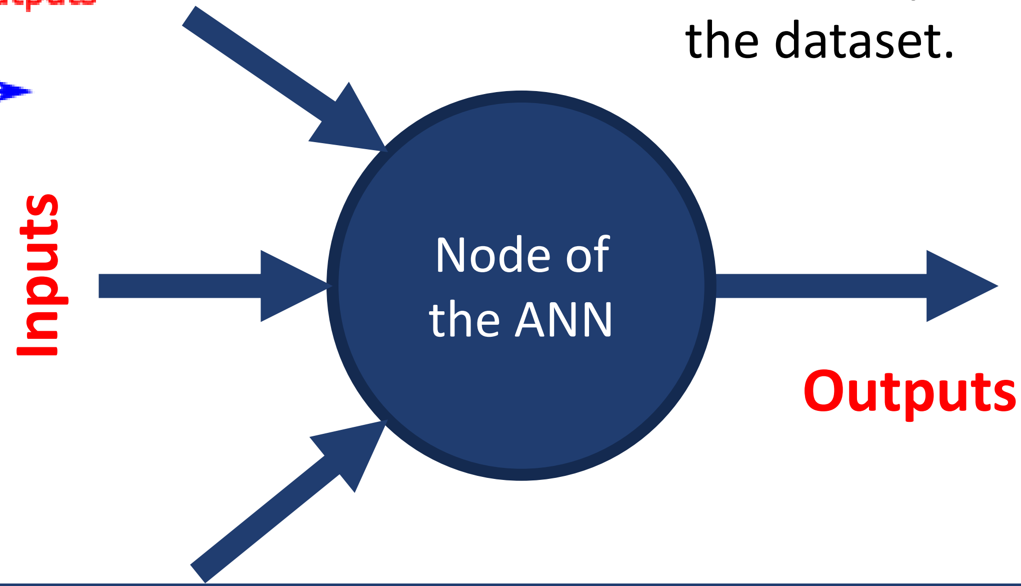
- Use biology as inspiration for mathematical model
- Get signals from previous neurons
- Generate signals (or not) according to inputs
- Pass signals on to next neurons
- By layering many neurons, can create complex models



# Neural Networks

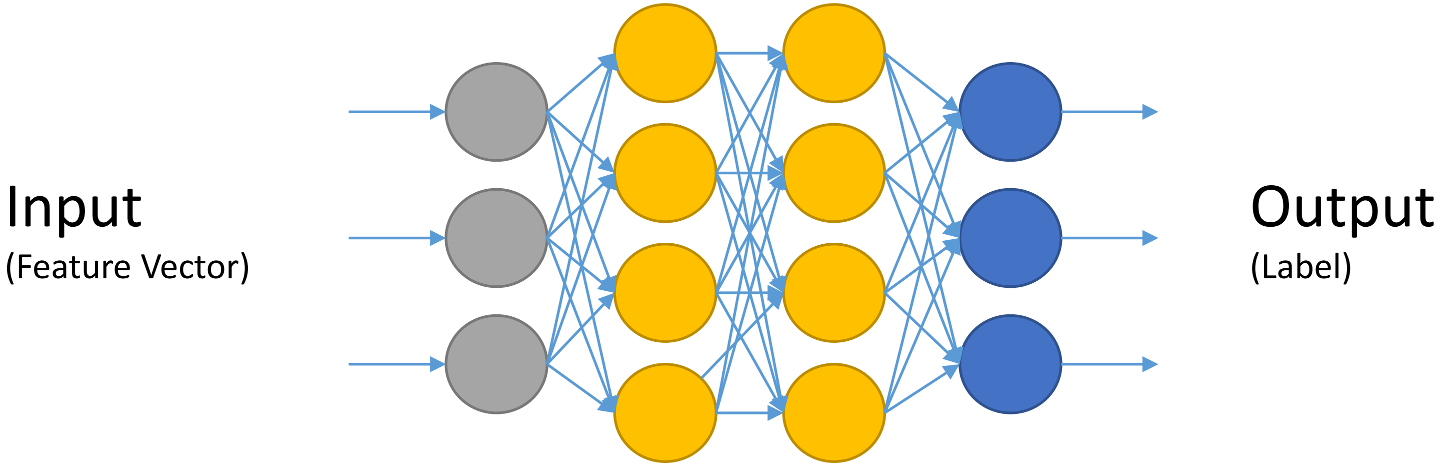



- We model neurons as a node, with input connections, and output connections.
- By having many neurons, layers, we can capture value hidden in the dataset.



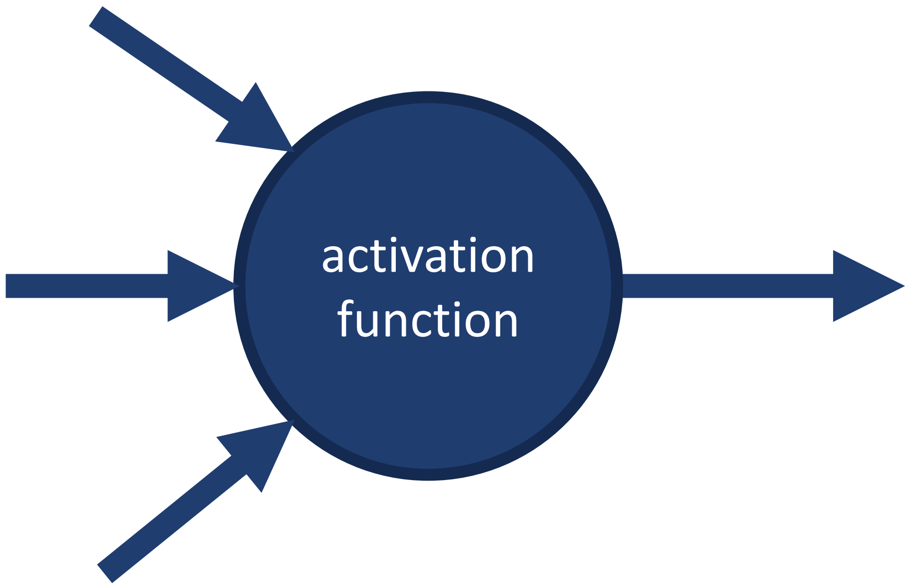

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

# Neural Network Structure



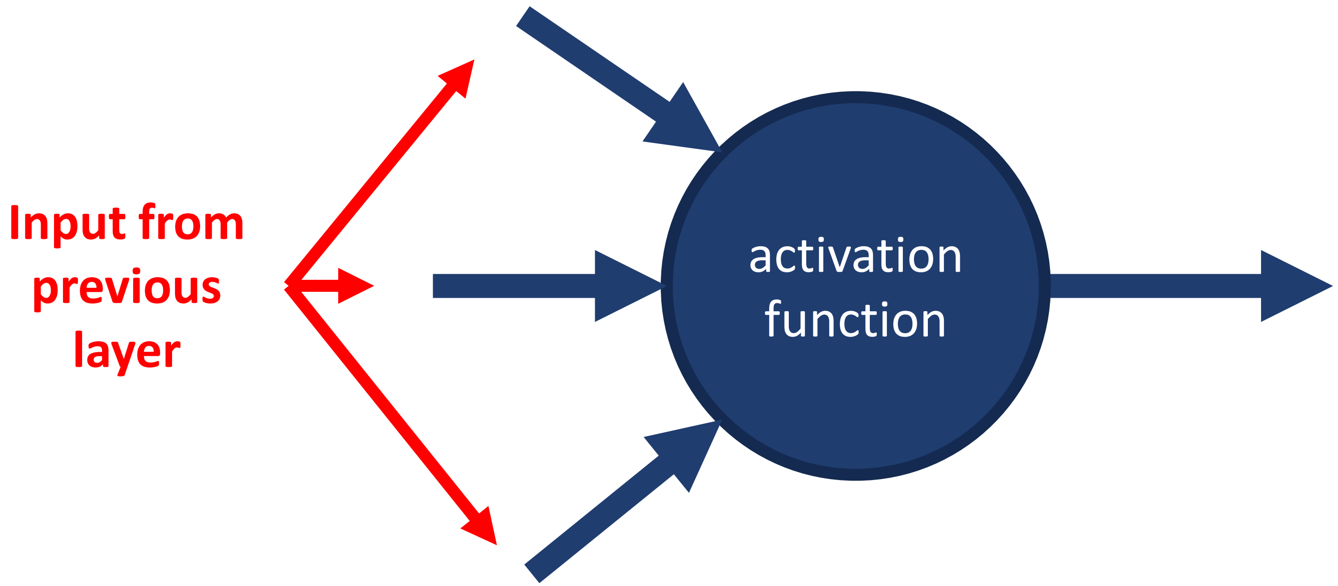

- Can think of it as a complicated computation engine
- We will ”train it” using our training data
- Then (hopefully) it will give good answers on new data

# Basic neuron visualization

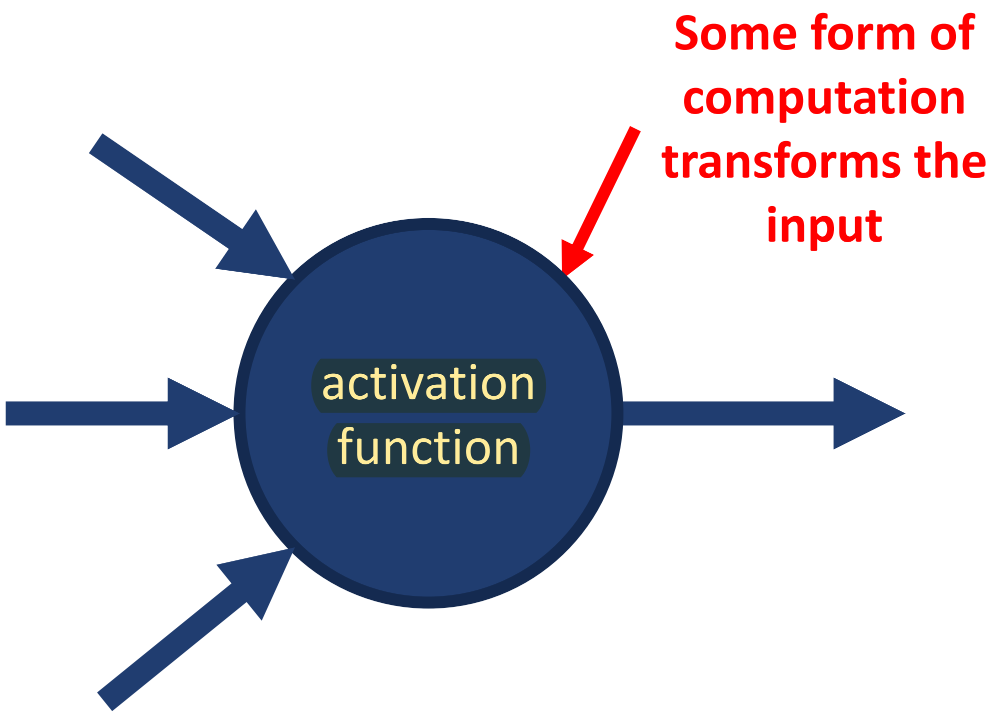



*Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021*

# Basic neuron visualization

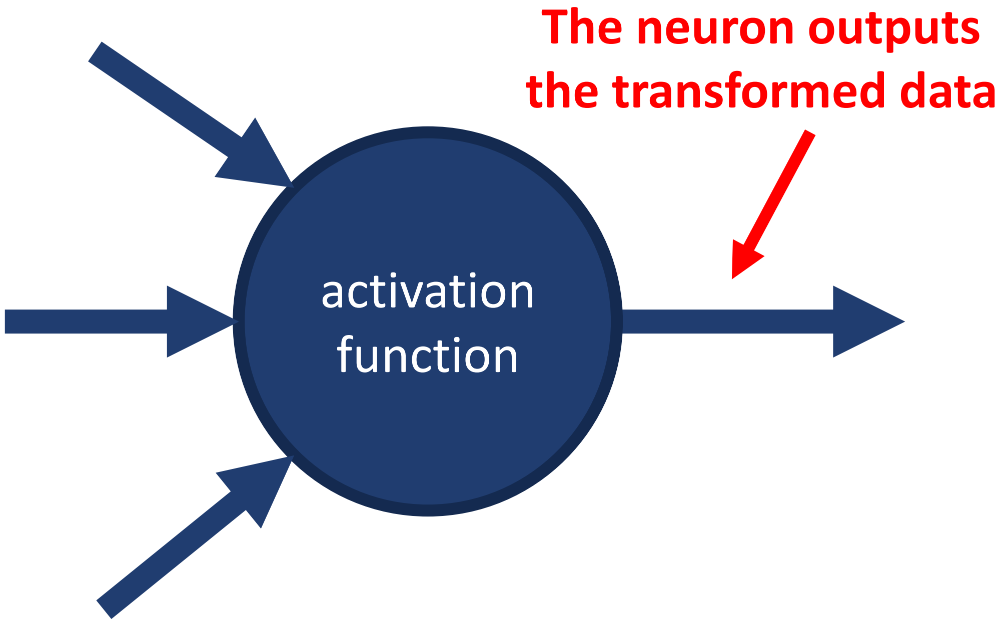


# Basic neuron visualization



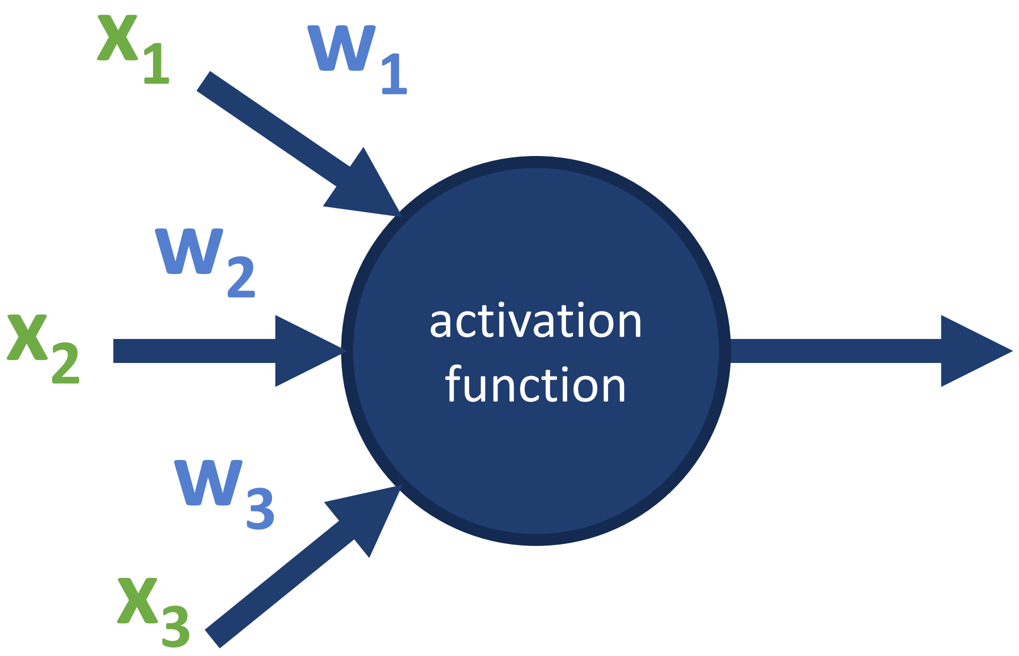
# Basic neuron visualization

- Note that there can be more than one connection to the next layer
- The activation function is the same for all of those.



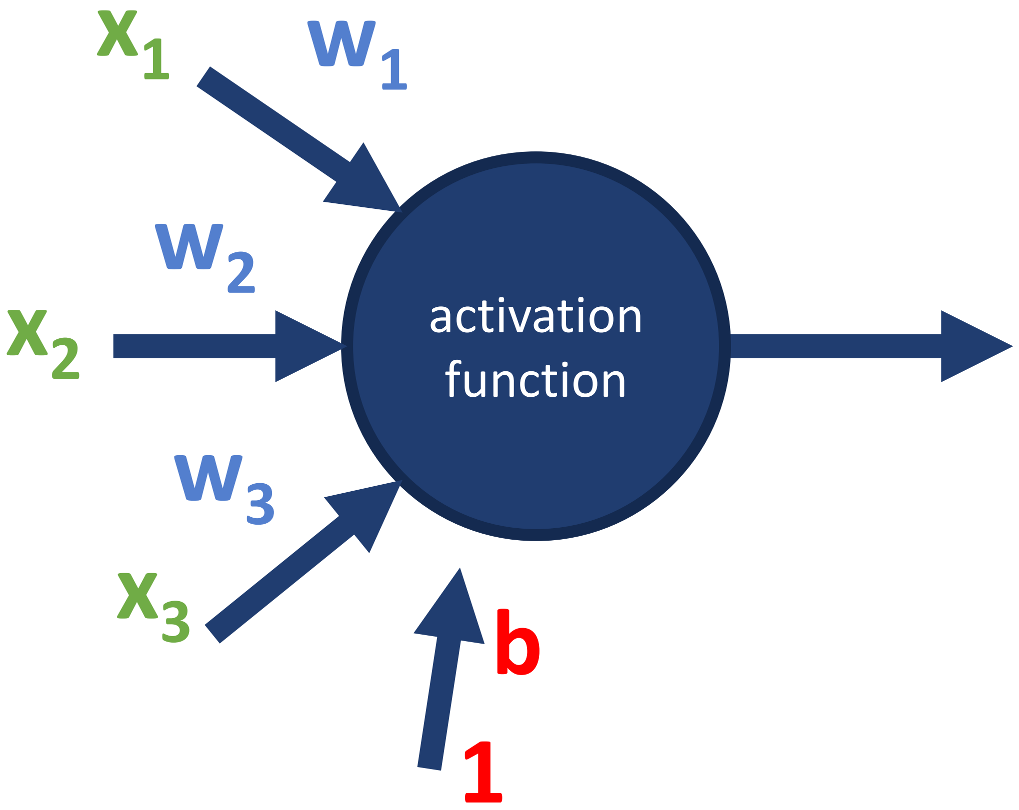
# Basic neuron visualization

- $x_1$, $x_2$, $x_3$ are inputs. They are real numbers.
- We have coefficients $w_1$, $w_2$, and $w_3$. They are real numbers. They form a row vector $[w_1, w_2, w_3]$

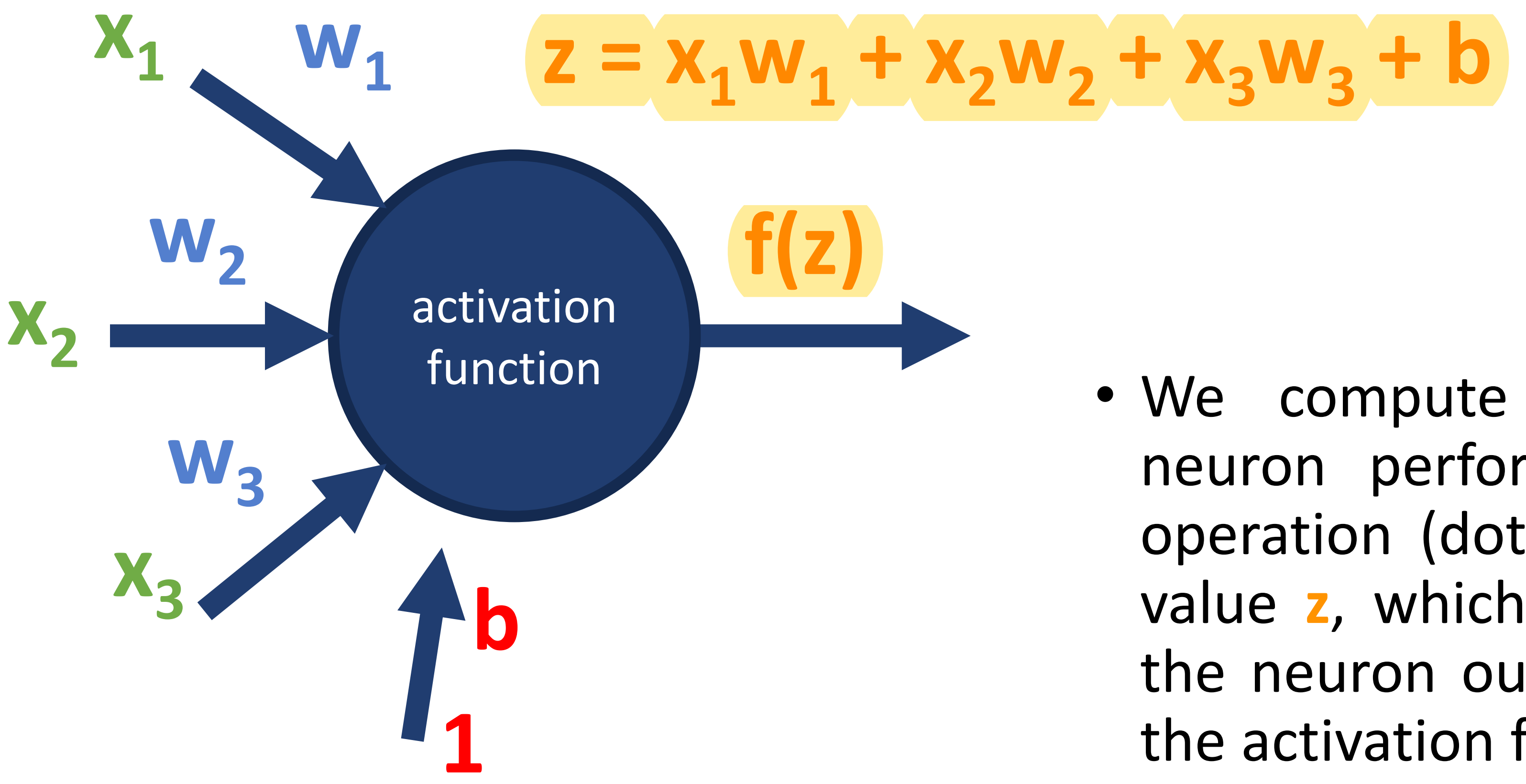


# Basic neuron visualization

- We can also add another input, called **bias term**.
- This is the analogue of intercept in Linear regression
- So our row vector of parameters is **[w1, w2, w3, b]**



# Basic neuron visualization




- We compute the value of the neuron performing linear algebra operation (dot product), and get a value z, which is used to compute the neuron outputs f(z), where f is the activation function.

# In vector notation

b = “bias term”

z = “net input”

f = activation function

a = output to next layer

$$z = b + \sum_{i=1}^{m} x_i w_i$$

$$z = b + x^T w$$

$$a = f(z)$$

Most of the operations are matrix multiplication, or element-wise operations on matrices and vectors

# Relation to logistic regression

- When we choose $f(z) = \frac{1}{1+e^{-z}}$

$$z = b + \sum_{i=1}^{m} x_i w_i = x_1 w_1 + x_2 w_2 + \cdots + x_m w_m + b$$

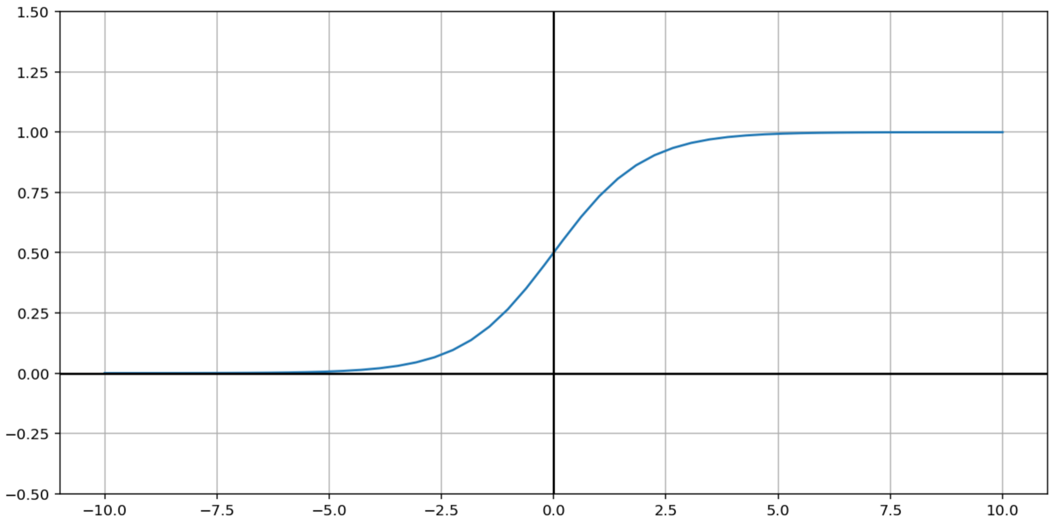
then a neuron is simply a ”unit” of logistic regression where

  - **weights** are coefficients
  - **inputs** are variables
  - **bias term** is the constant term

- This neuron is doing logistic regression on the input

# Relation to Logistic Regression

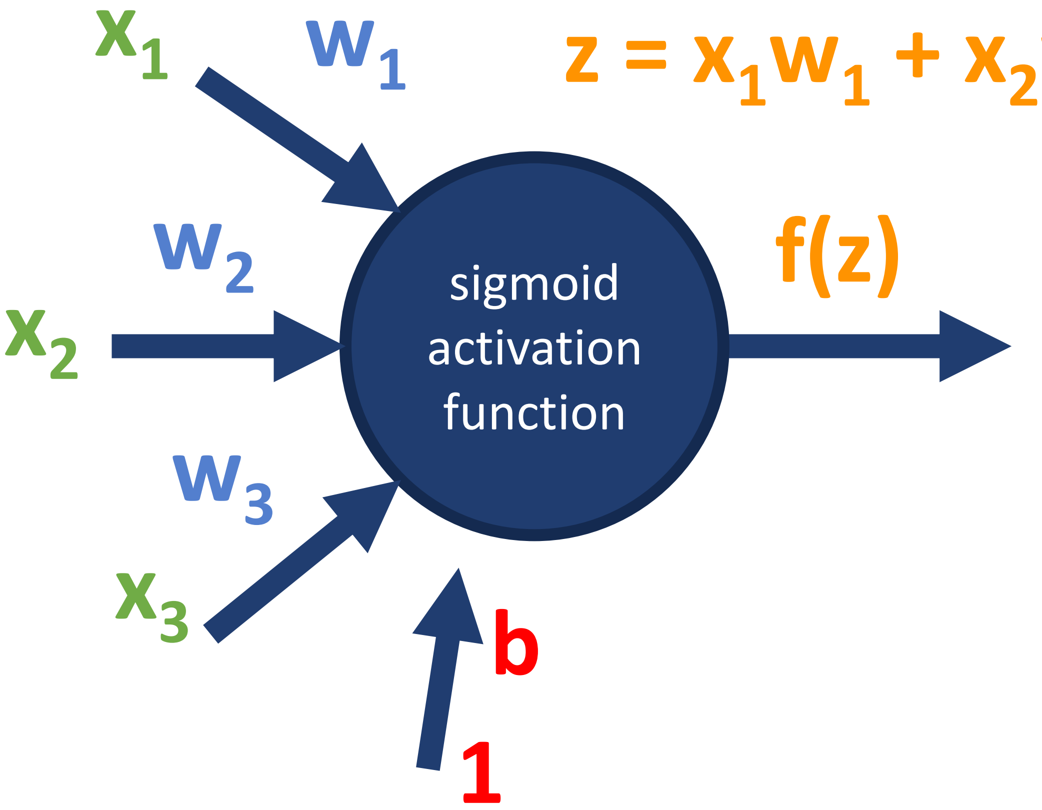
- This is called the "sigmoid" function $\sigma(z) = \frac{1}{1+e^{-z}}$



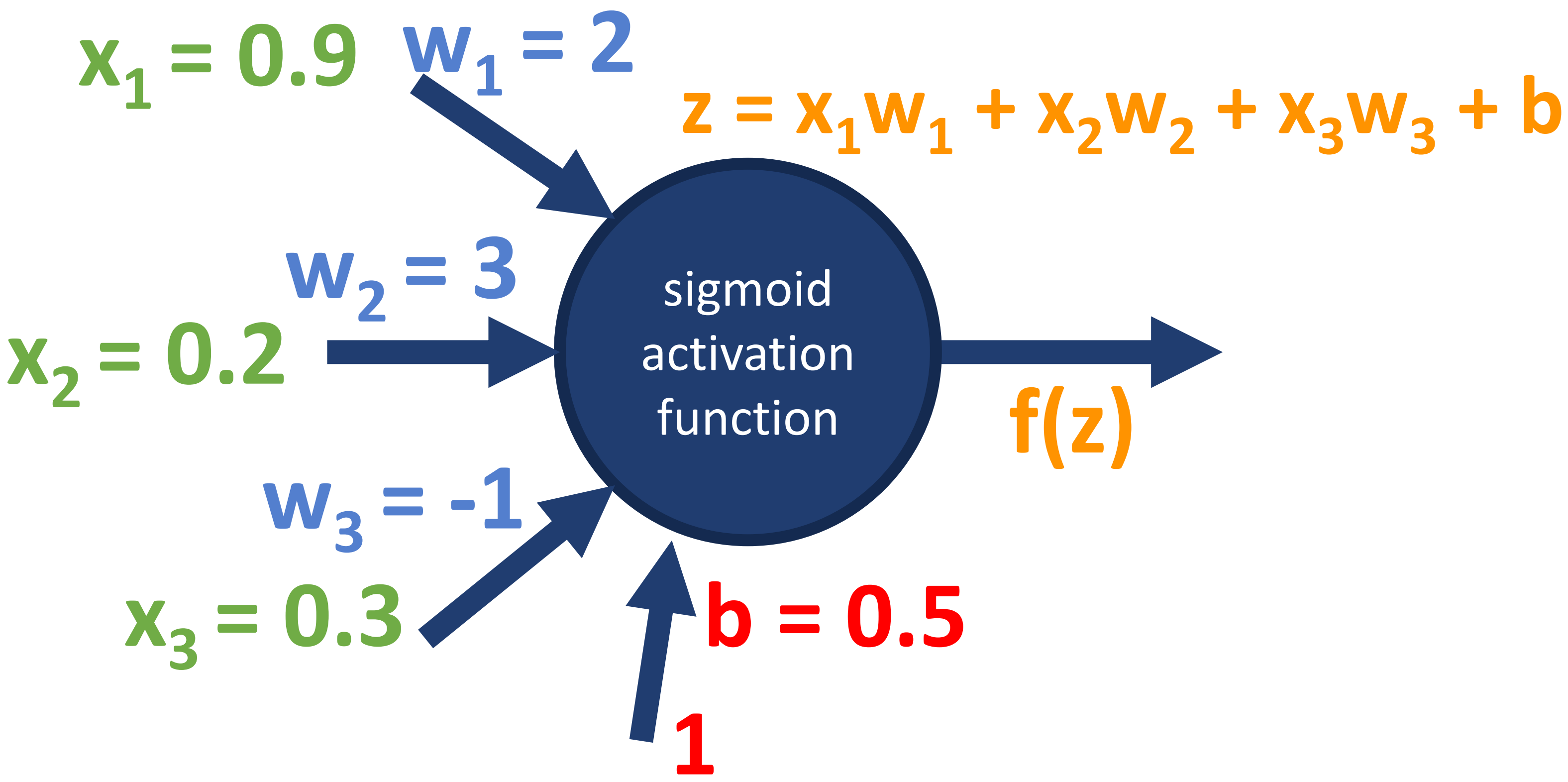

- There are many other activation functions. We will view some of them later on.

# Example of Neuron Computation

$$z = x_1w_1 + x_2w_2 + x_3w_3 + b$$



# Example of Neuron Computation

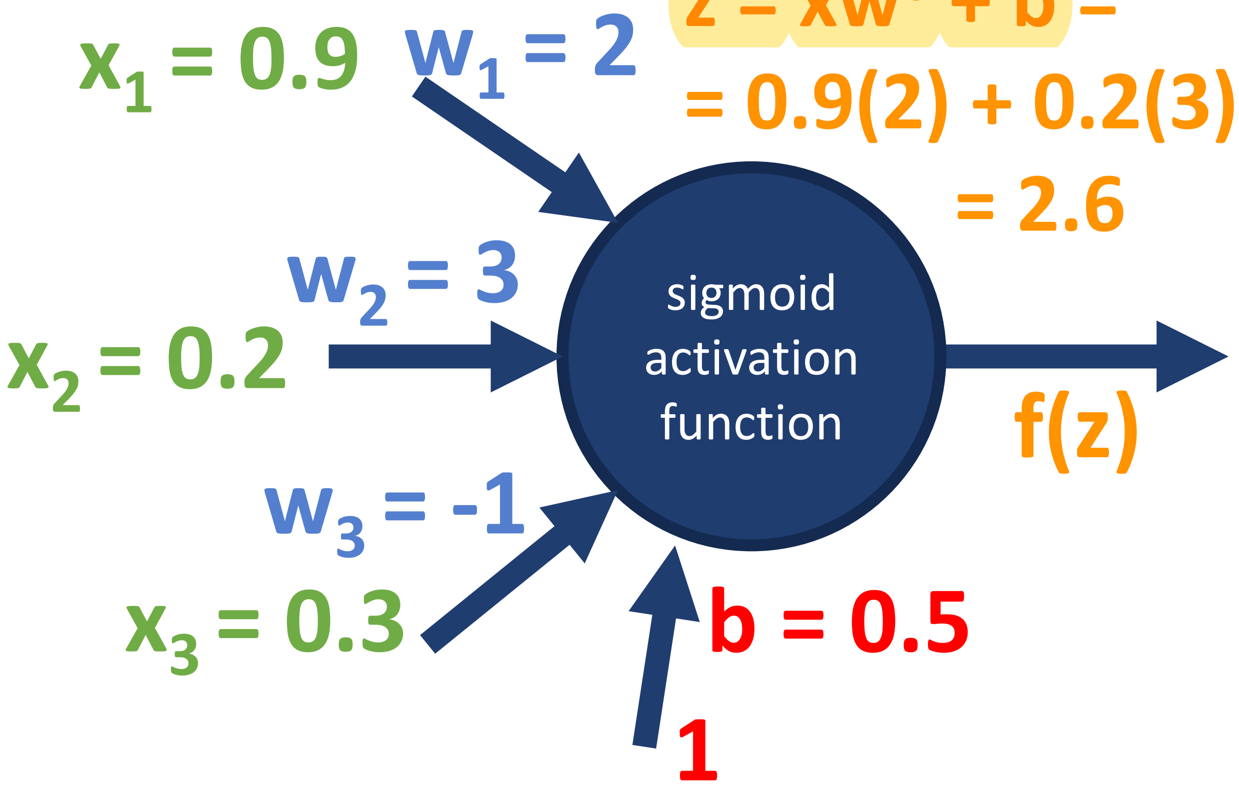



- The weights are [2,3,-1], the bias term 0.5, the inputs [0.9,0.2,0.3].

*Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021*

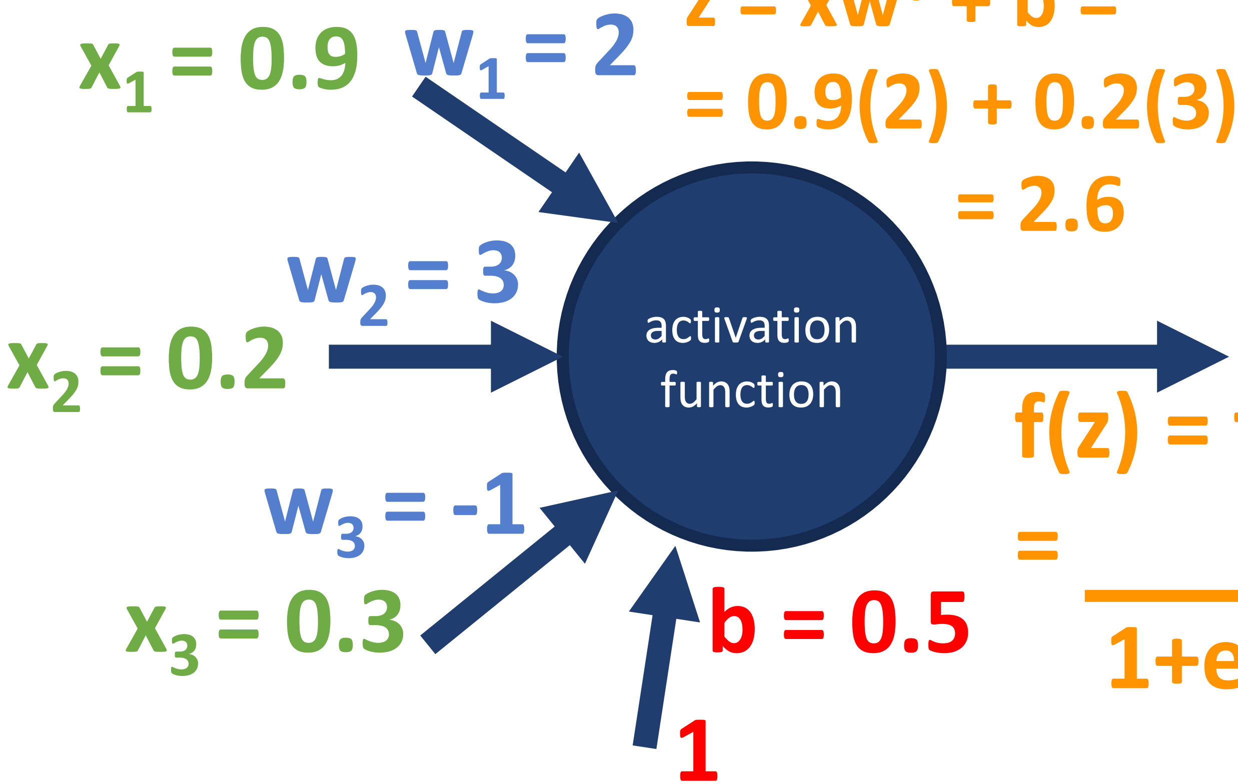
# Example of Neuron Computation

$$z = xw^T + b =$$
$$= 0.9(2) + 0.2(3) + 0.3(1) + 0.5$$
$$= 2.6$$




- The weights are [2,3,-1], the bias term 0.5, the inputs [0.9,0.2,0.3].

# Example of Neuron Computation




$$z = xw^T + b =$$
$$= 0.9(2) + 0.2(3) + 0.3(1) + 0.5$$
$$= 2.6$$

$$f(z) = f(2.6) =$$
$$= \frac{1}{1+\exp(-2.6)}$$

This neuron would output the value **0.93**

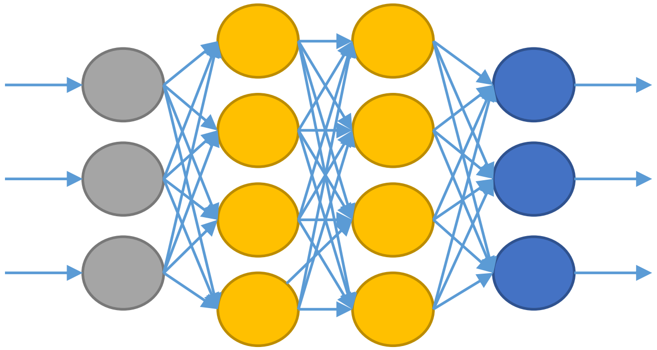
*Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021*

Regione Emilia-Romagna

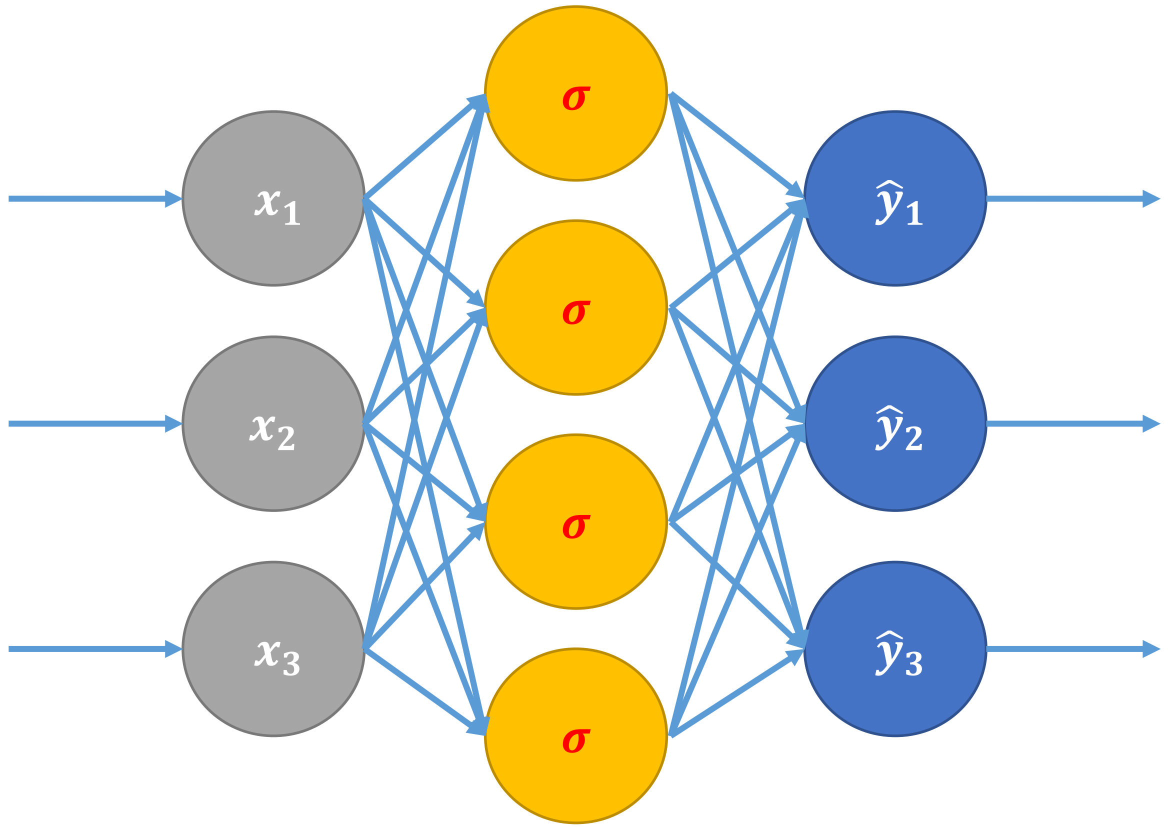
Università degli Studi di Ferrara

# Why neural nets?

- Deeper structures allow the build-up of more "intermediate" results that are then used to make the final prediction
- The layers in between act like "feature generation" for the final layer, which predicts the output.

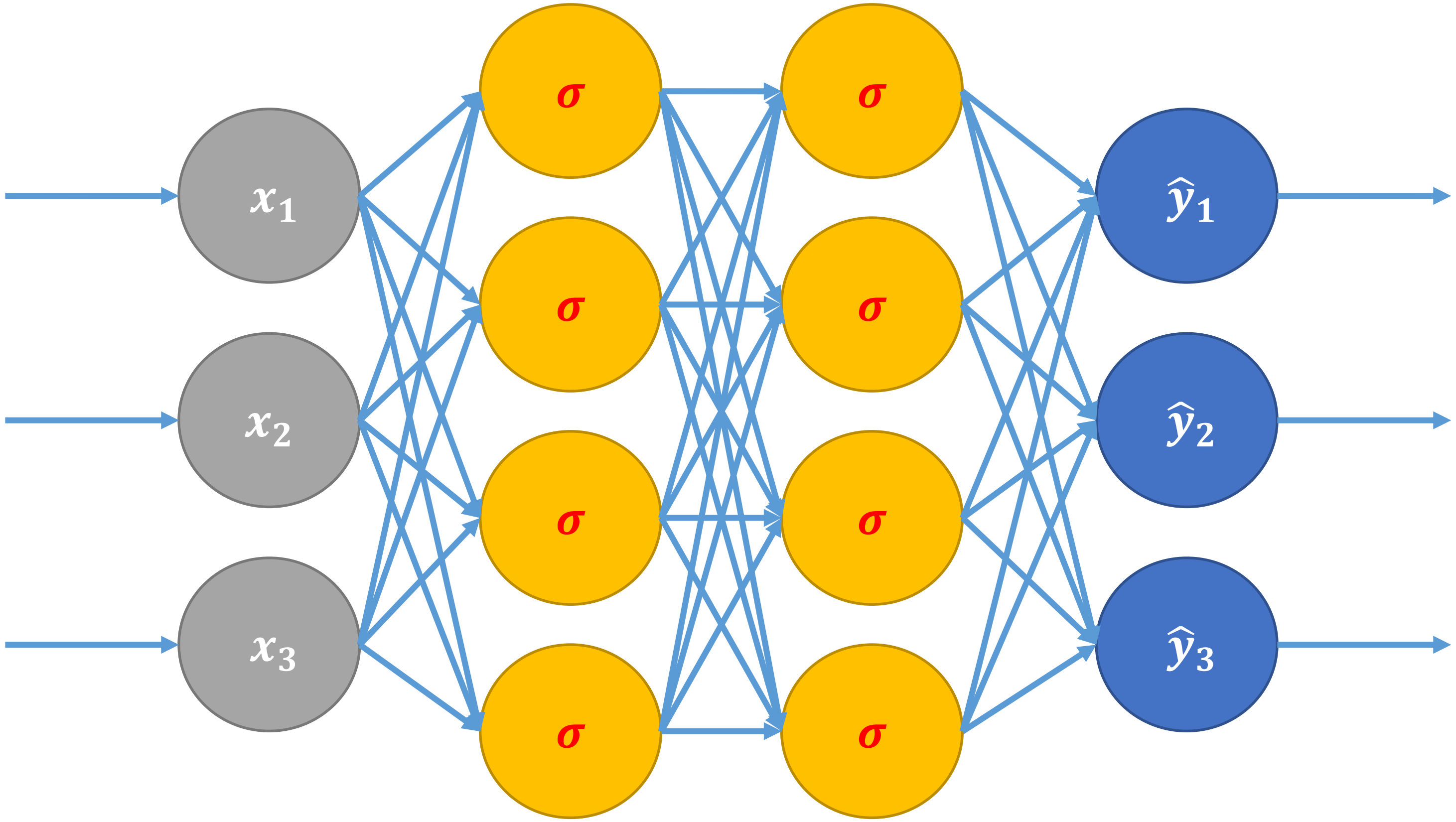


# Shallow Neural Network

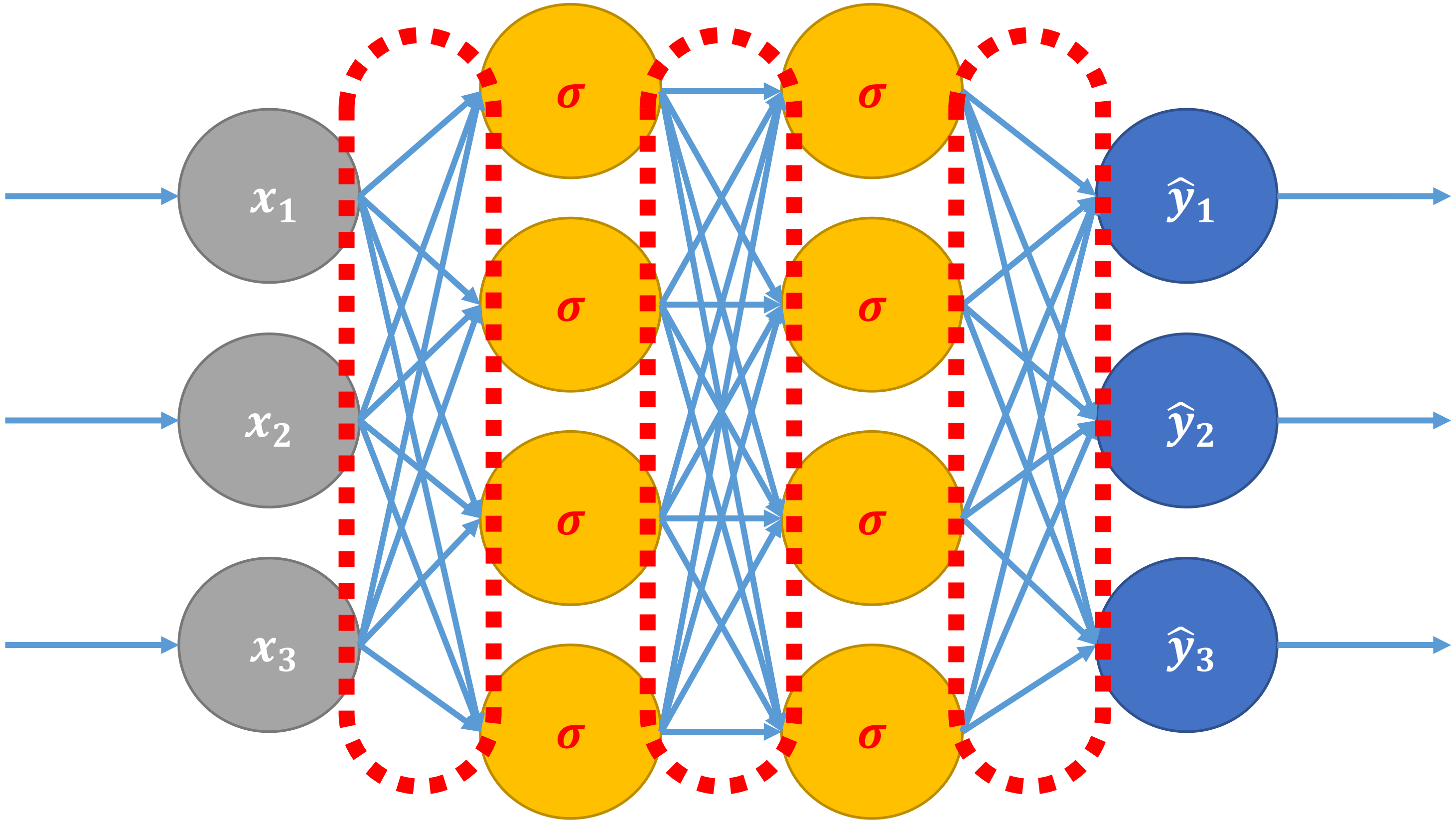



*Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021*

# Deep Neural Network

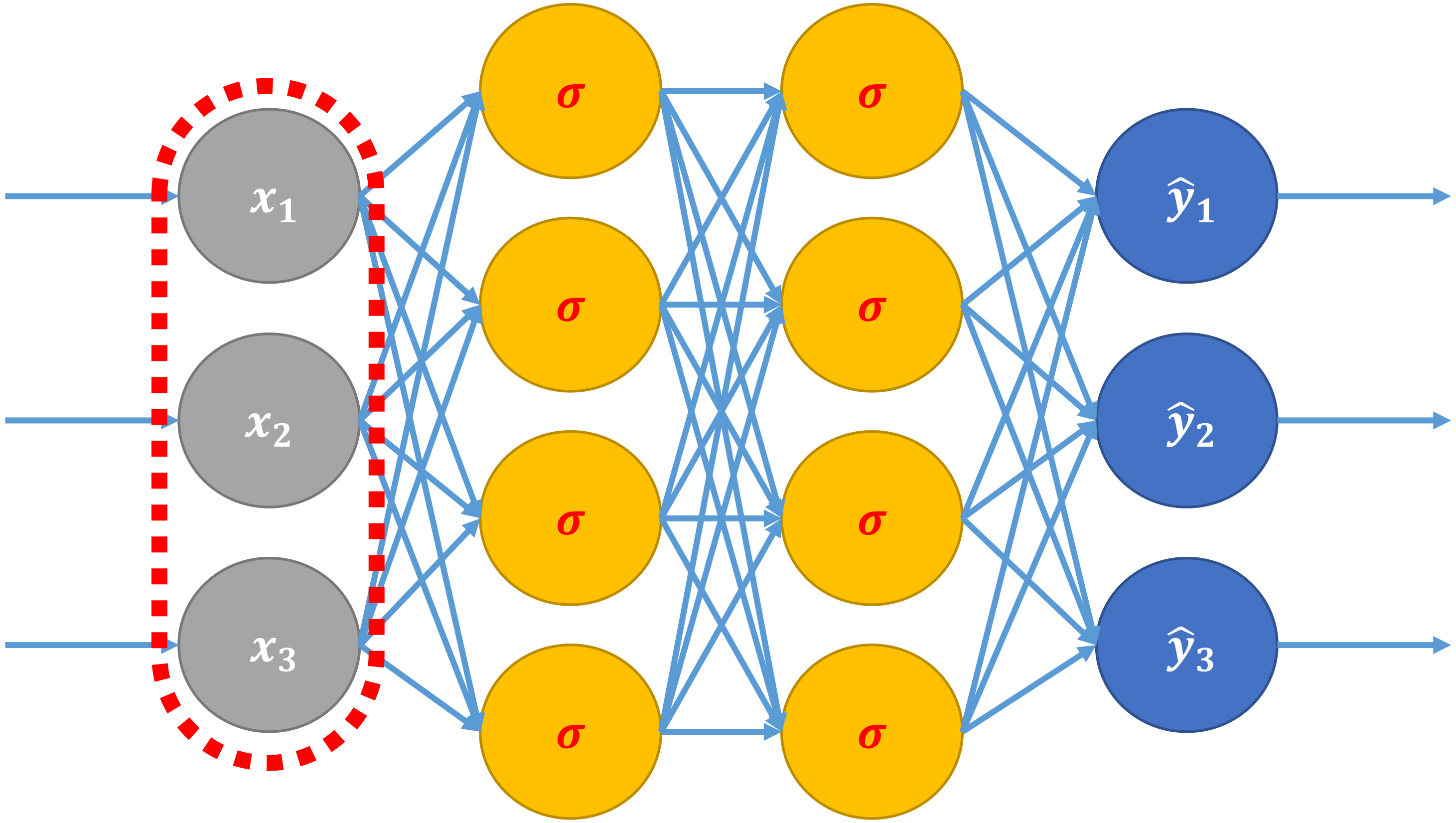


# Deep Neural Network



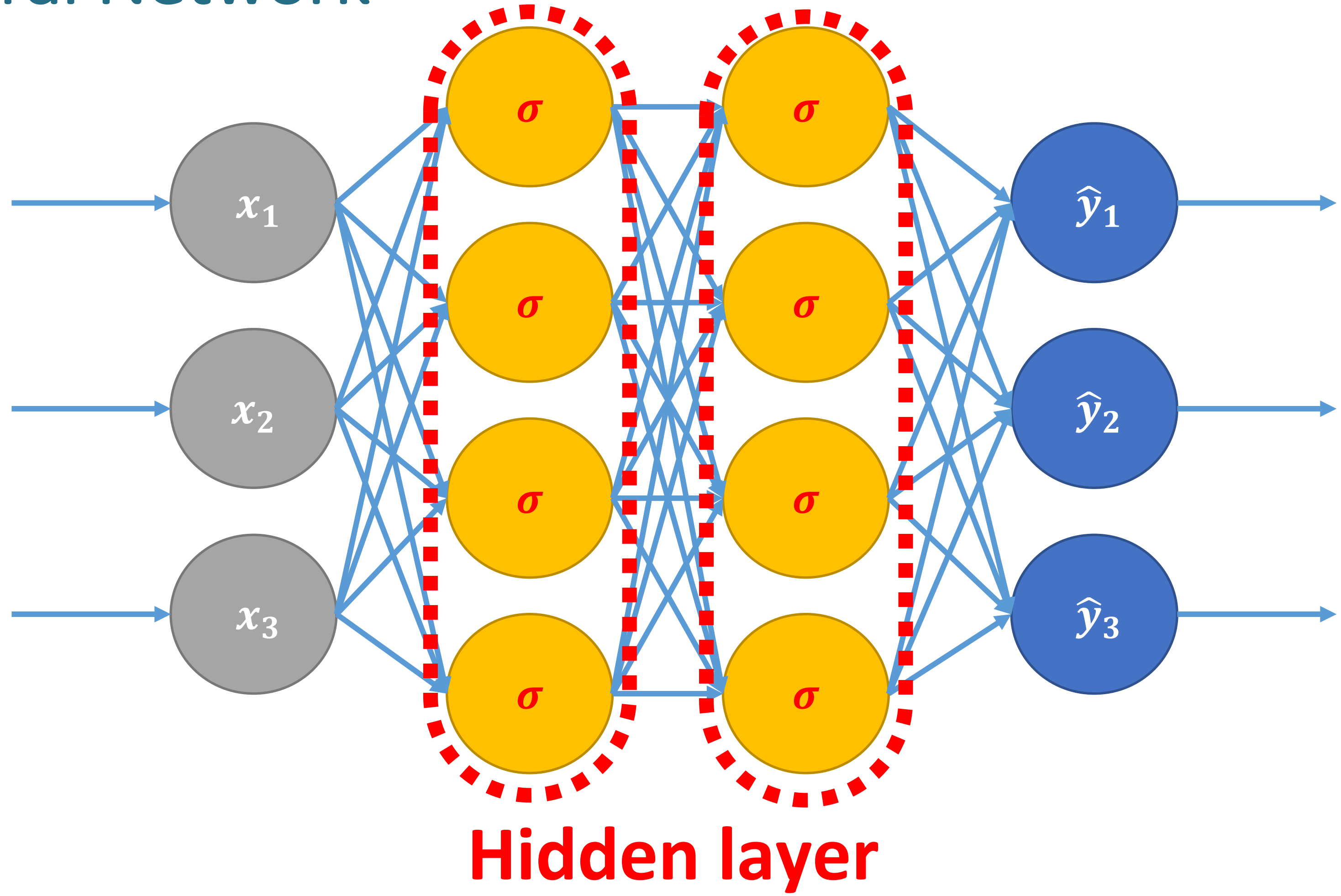

**Weights**

# Deep Neural Network

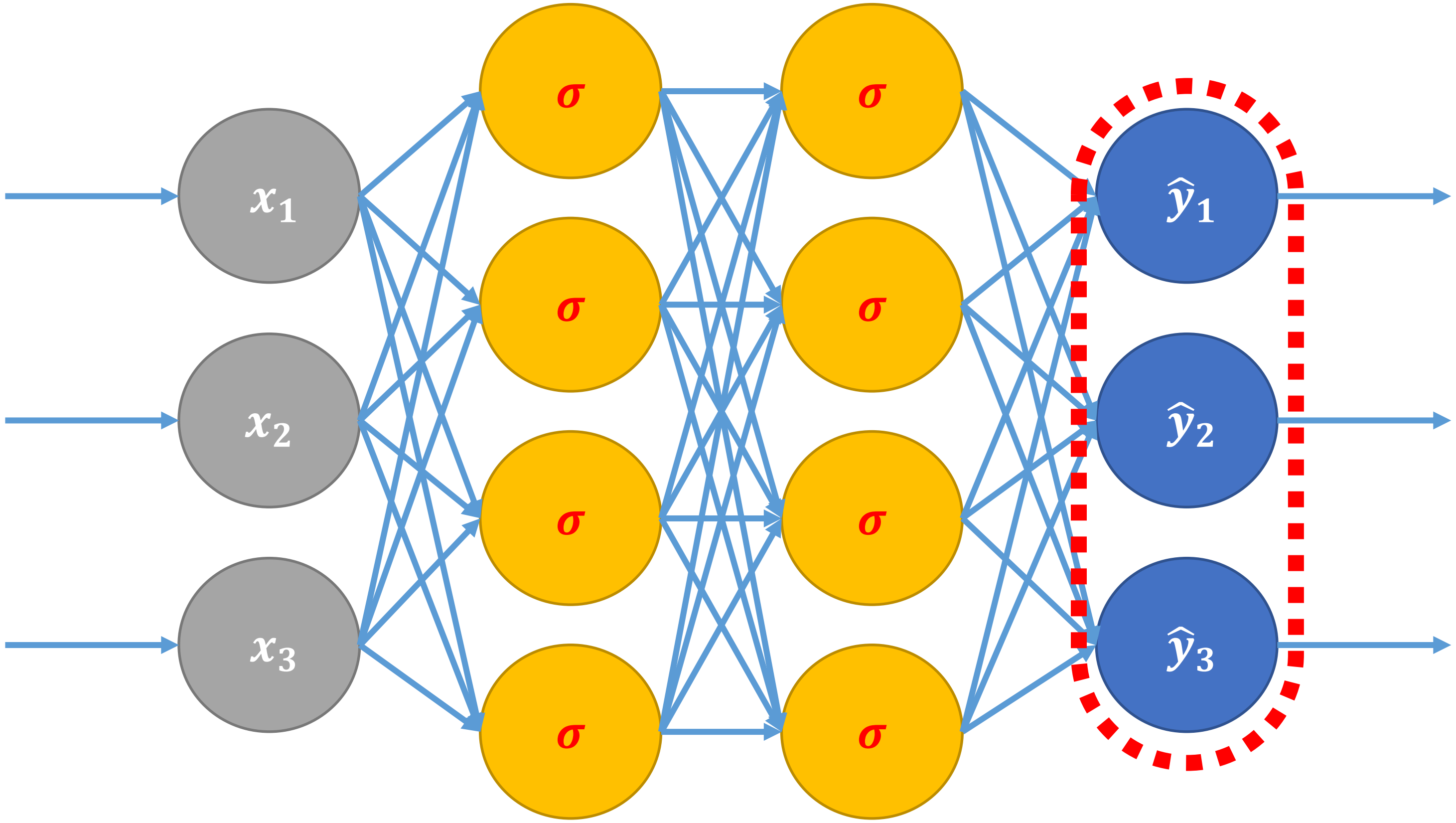



**Input layer**

# Deep Neural Network



# Deep Neural Network

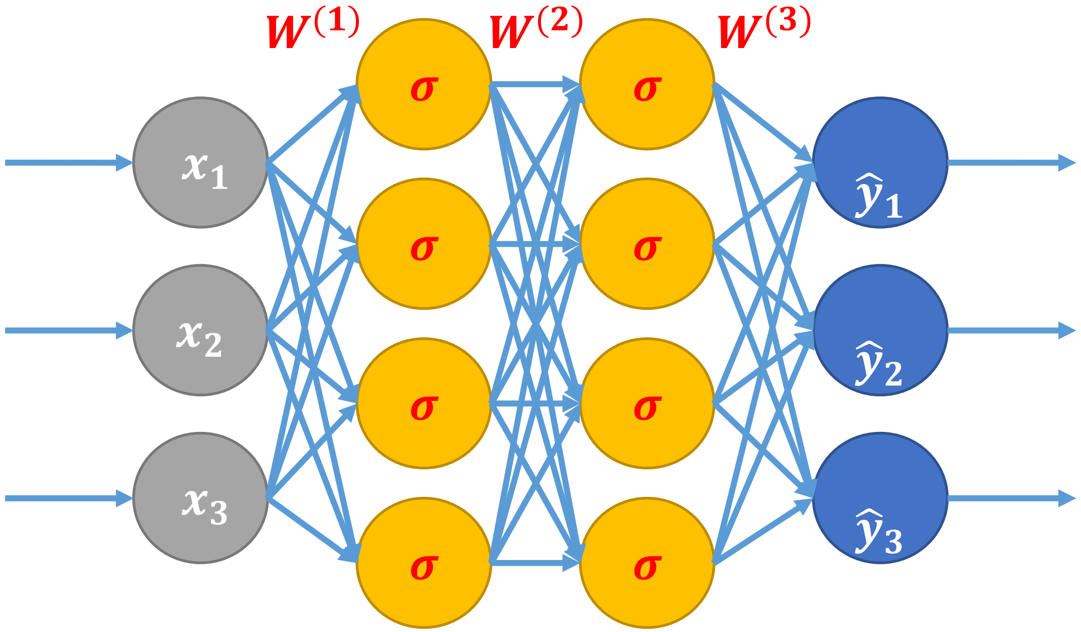



**Output layer**

*Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021*

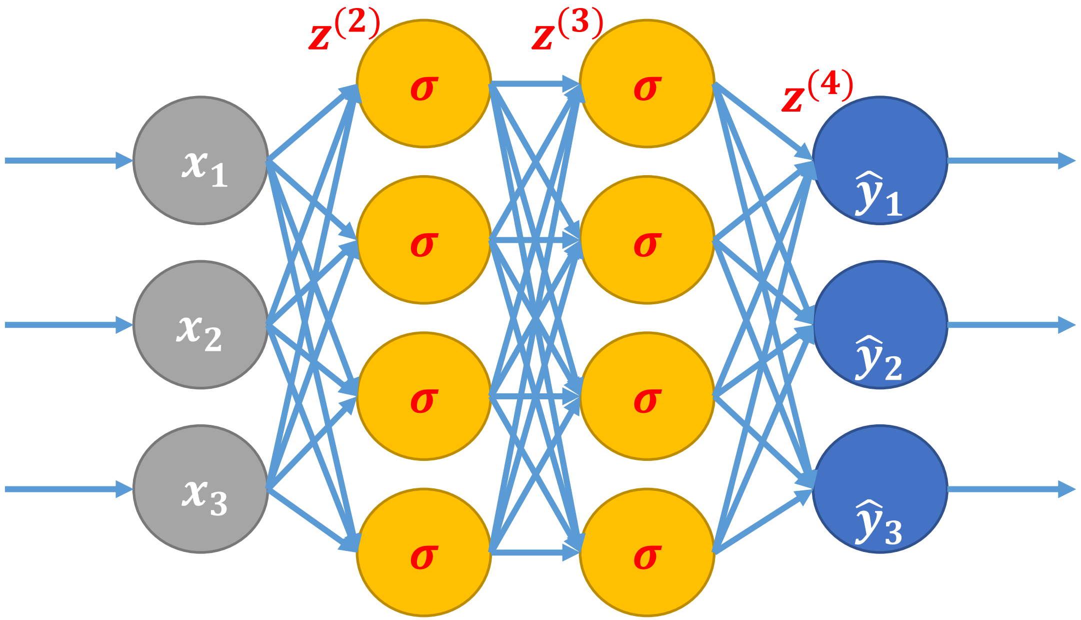
# Deep Neural Network

- **Weights** between layers can be represented as a **matrix**, by putting every incoming weight to a given node as a row vector, and stacking those rows.
- So, between the input layer and the first hidden layer, we have a 3x4 matrix.
- Note that bias is not considered here.



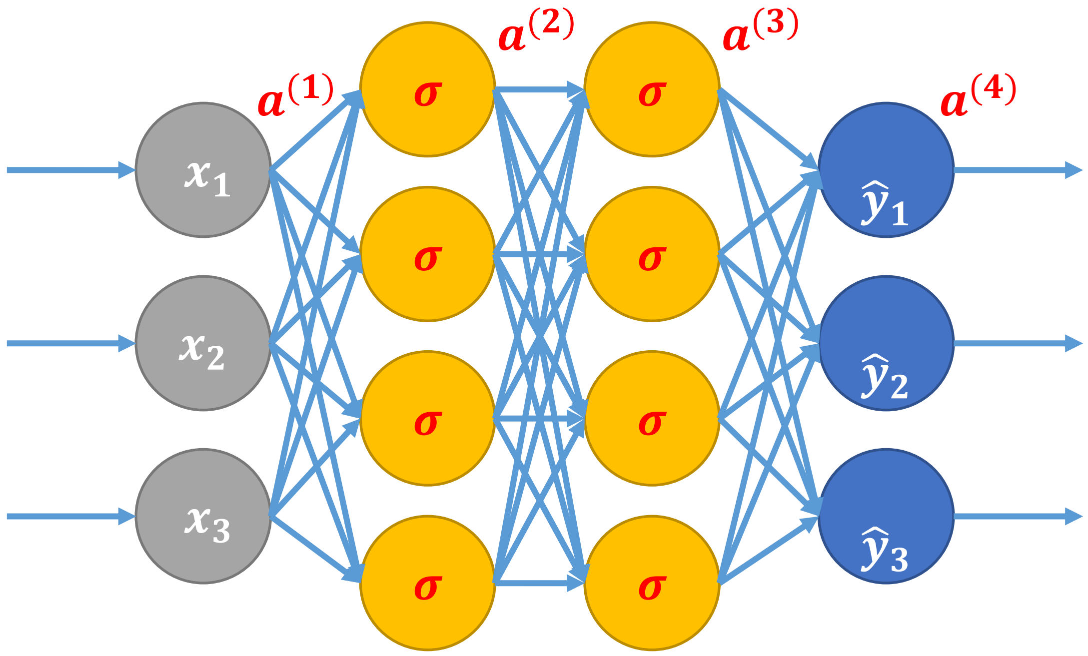
# Deep Neural Network

- Once we have the weights, then we can compute the **net inputs**
- It just corresponds to a **matrix multiplication**.

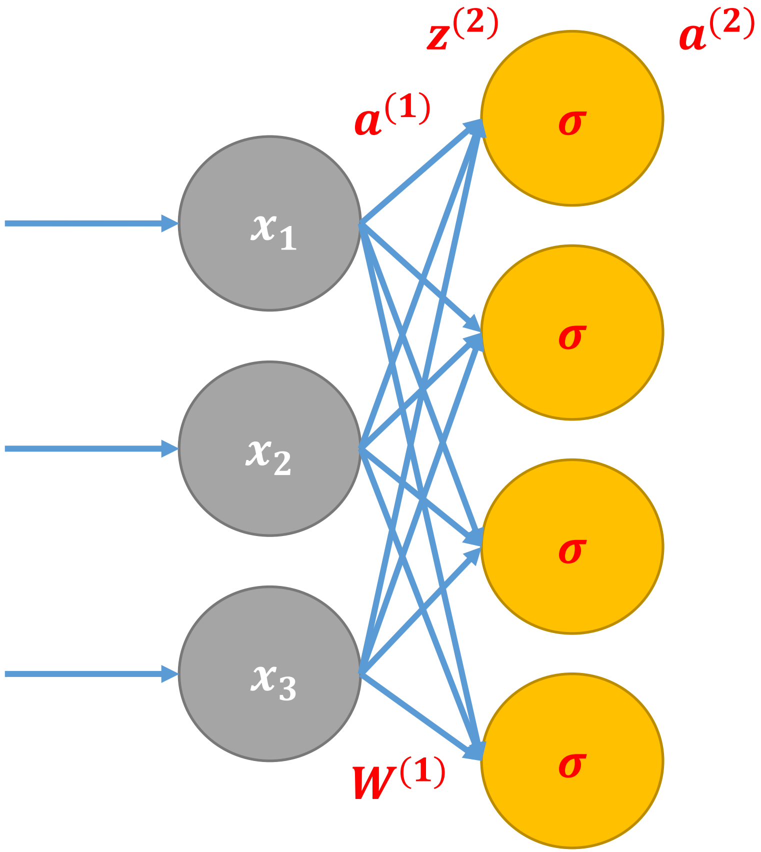


# Deep Neural Network

- Once the net input is prepared, it is fed into the **activation function**, and passed as input to the next layer.



## Deep Neural Network




$$x = a^{(1)}$$

$$z^{(2)} = xW^{(1)}$$

$$a^{(2)} = \sigma\left(z^{(2)}\right)$$

$a^{(1)}$ is a 3-row vector
$W^{(1)}$ is a 3x4 matrix
$z^{(2)}$ is a 4-row vector
$a^{(2)}$ is a 4-row vector

*Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021*

# Continuing the computation

- For a single training instance (data point)
  - Input: vector $x$ (a row vector of length 3)
  - Output: vector $\hat{y}$ (a row vector of length 3)

$$z^{(2)} = xW^{(1)} \qquad a^{(2)} = \sigma(z^{(2)})$$
$$z^{(3)} = a^{(2)}W^{(2)} \qquad a^{(3)} = \sigma(z^{(3)})$$
$$z^{(4)} = a^{(3)}W^{(3)} \qquad \hat{y} = softmax(z^{(4)})$$

**Softmax** is another activation function.

# Continuing the computation

- In practice, we do these computation for many data points at the same time, by "stacking" the rows into a matrix.
  - Input: matrix x (a nx3 matrix) (each row a single instance)
  - Output: vector $\hat{y}$ (a nx3 matrix) (each row a single prediction)

$$
\begin{aligned}
z^{(2)} &= xW^{(1)} & a^{(2)} &= \sigma(z^{(2)}) \\
z^{(3)} &= a^{(2)}W^{(2)} & a^{(3)} &= \sigma(z^{(3)}) \\
z^{(4)} &= a^{(3)}W^{(3)} & \hat{y} &= softmax(z^{(4)})
\end{aligned}
$$

- Equations look the same!