MACHINE LEARNING

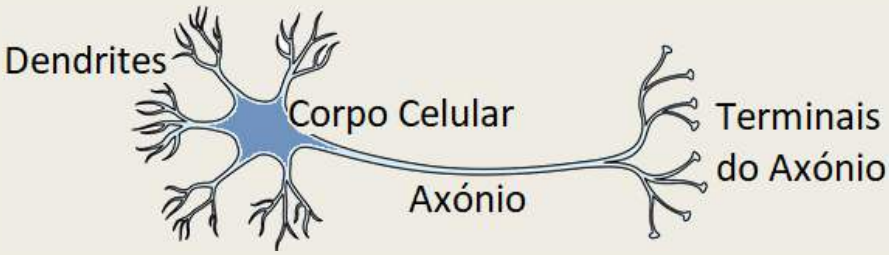
With content adapted from the thesis PhD “Desenvolvimento de Modelos Analíticos de Apoio à Gestão de Instituições do Ensino Superior, com Recurso a Data Mining”, of Maria Martins, 2020

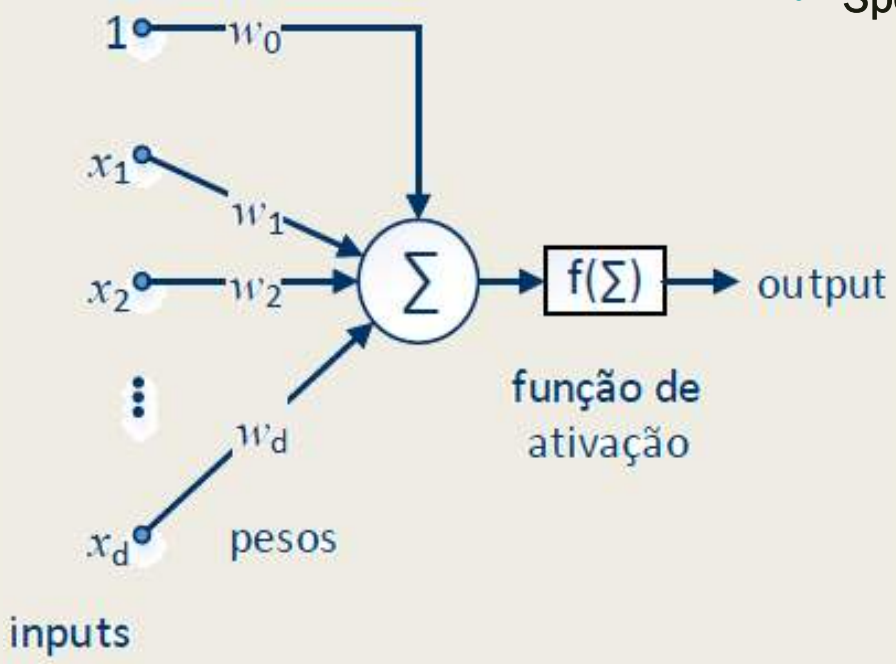
Neural Networks

- An (artificial) Neural Network is a learning technique inspired by biological intelligence,
  - *It is based on the structure and operation of the nervous system, to simulate the learning capacity of the human brain in the acquisition of knowledge*
  - *It is an extremely interconnected structure of computational units called artificial neurons, with the capacity to learn*
    - thus representing simplified approximations of the networks of neurons that are found in the human brain
- It's one of the most powerful and widely used machine learning methods.

The Neuron

- The main processing component of a Neural Network (RN) is the neuron
  - *These units, densely interconnected through a pattern of connections, play a very simple role that consists of receiving signals from the input connections and them calculating a new value to be sent to the output*





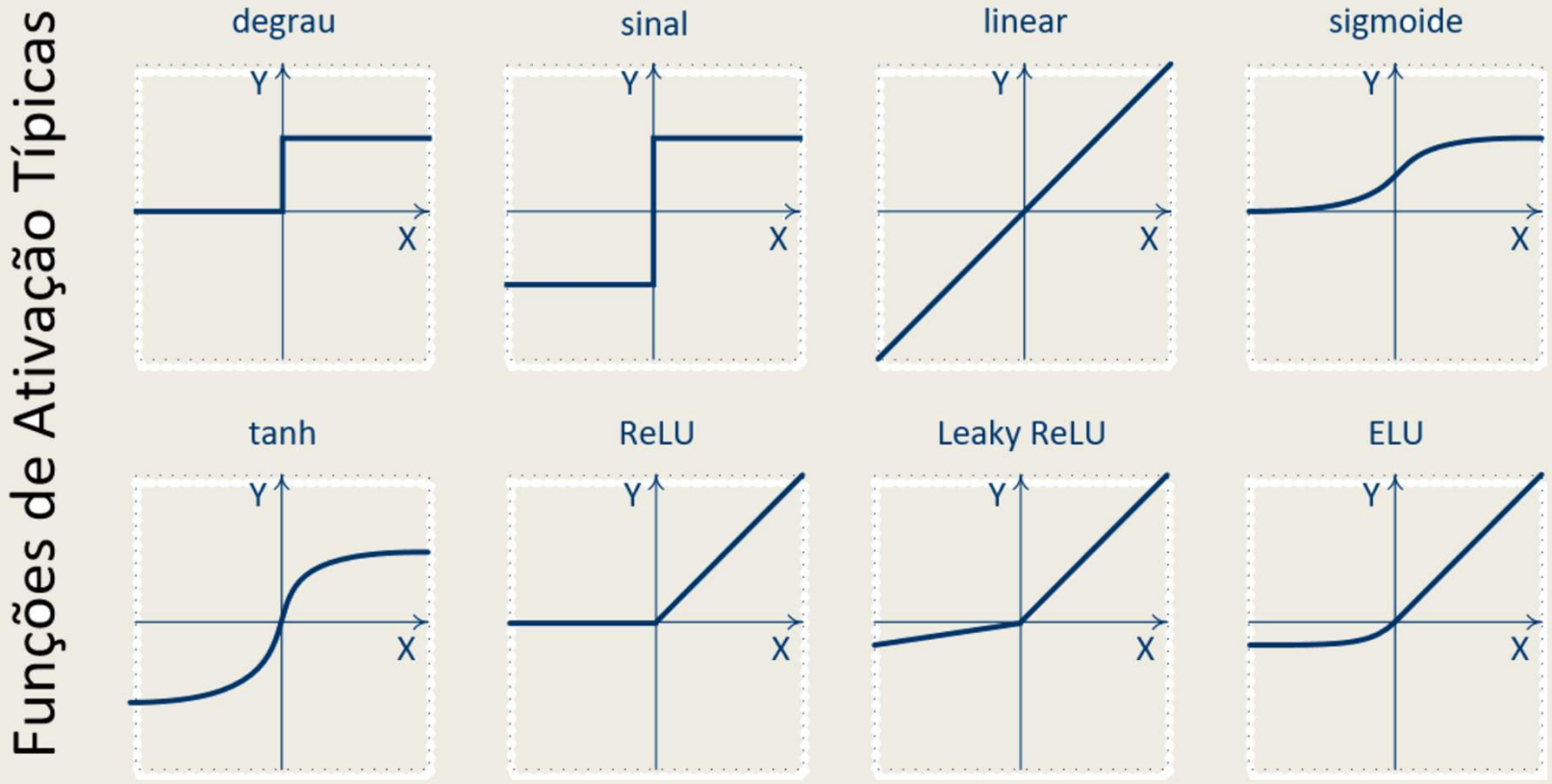
- Specifically,
  - *Each input terminal of the neuron, simulating a dendrite, receives a value*
  - *The values received are then weighted and added together*
  - *Being then the resulting value, added of an offset value (or bias – the W_0), used to calculate the output value of the neuron, in analogy with the processing performed by the cell body or sum*
  - *A specific function performs this calculation, called an **activation function**, which typically takes one of the typical forms illustrated at left.*

- The input of the activation function can then be expressed simply by the following expression:

$$s = \mathbf{w}^t \cdot \mathbf{x} + w_0 = \sum_{i=1}^{d} w_i x_i + w_0$$

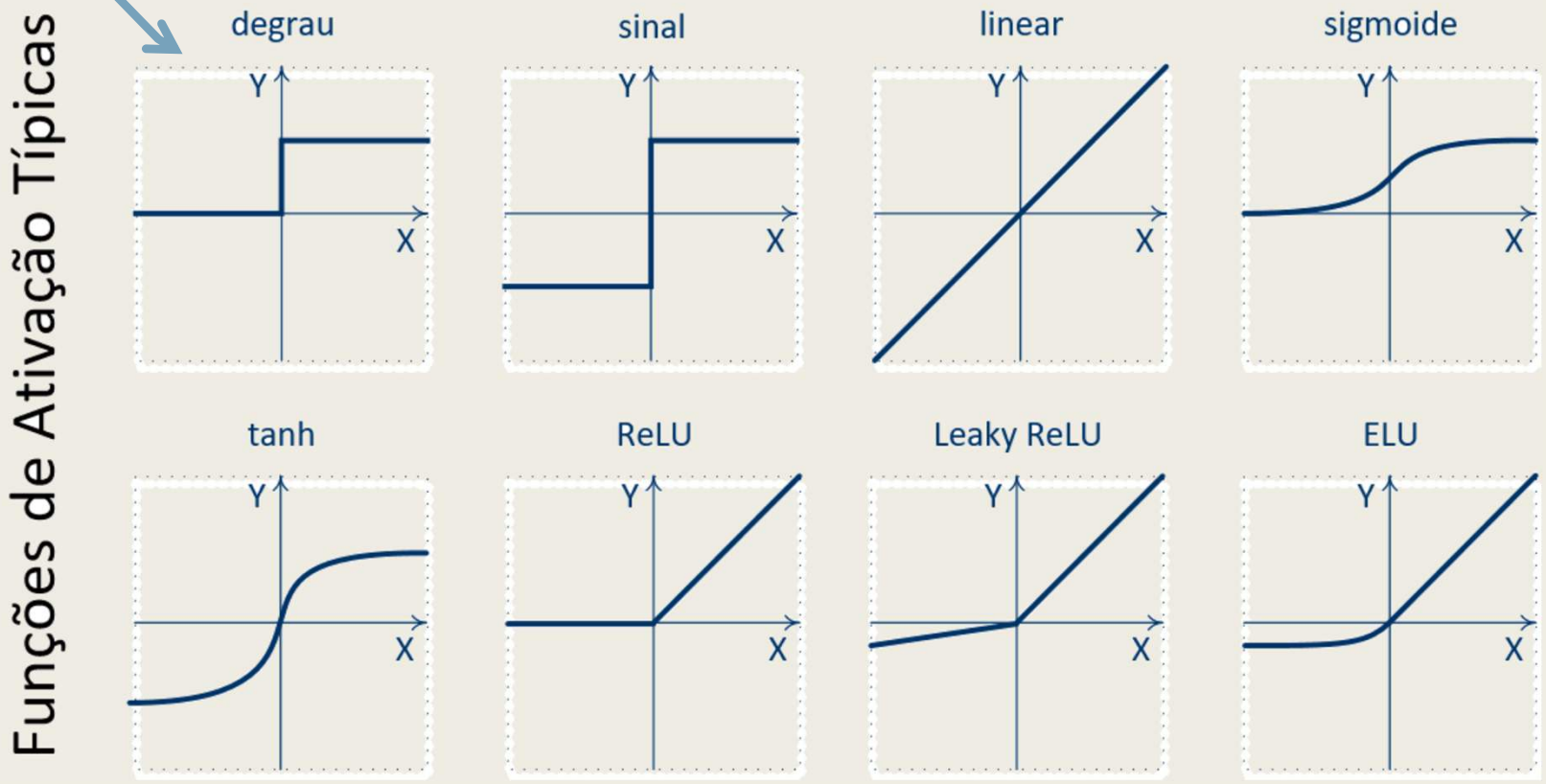
where $x = [x_1, x_2, x_3, \ldots, x_d]^t$ *is the array with the set of D inputs of the neuron,* $w = [w_1, w_2, w_3, \ldots, w_d]^t$ *the array of associated weights, and* w_0 *the value of offset or bias.*

  - *It is these weights, associated with the inputs of neurons, that constitute the main means of storing information in a network.*



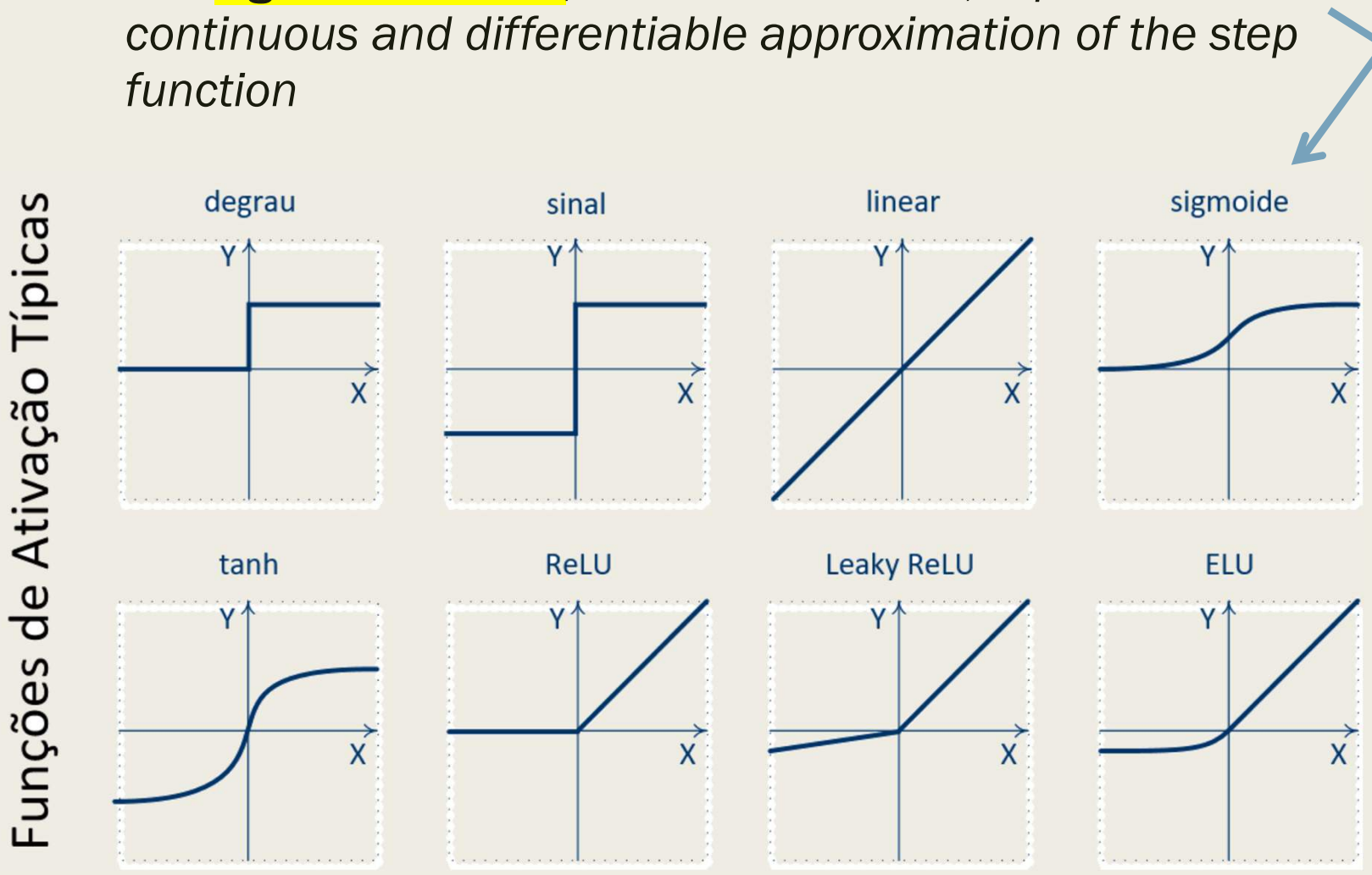
Activation Functions

- The output of the neuron will then depend on the activation function that is chosen:
  - *With step function, The signal will be 0 as long as the input S is negative (neuron activated), becoming 1 as soon as it becomes positive (neuron activated)*



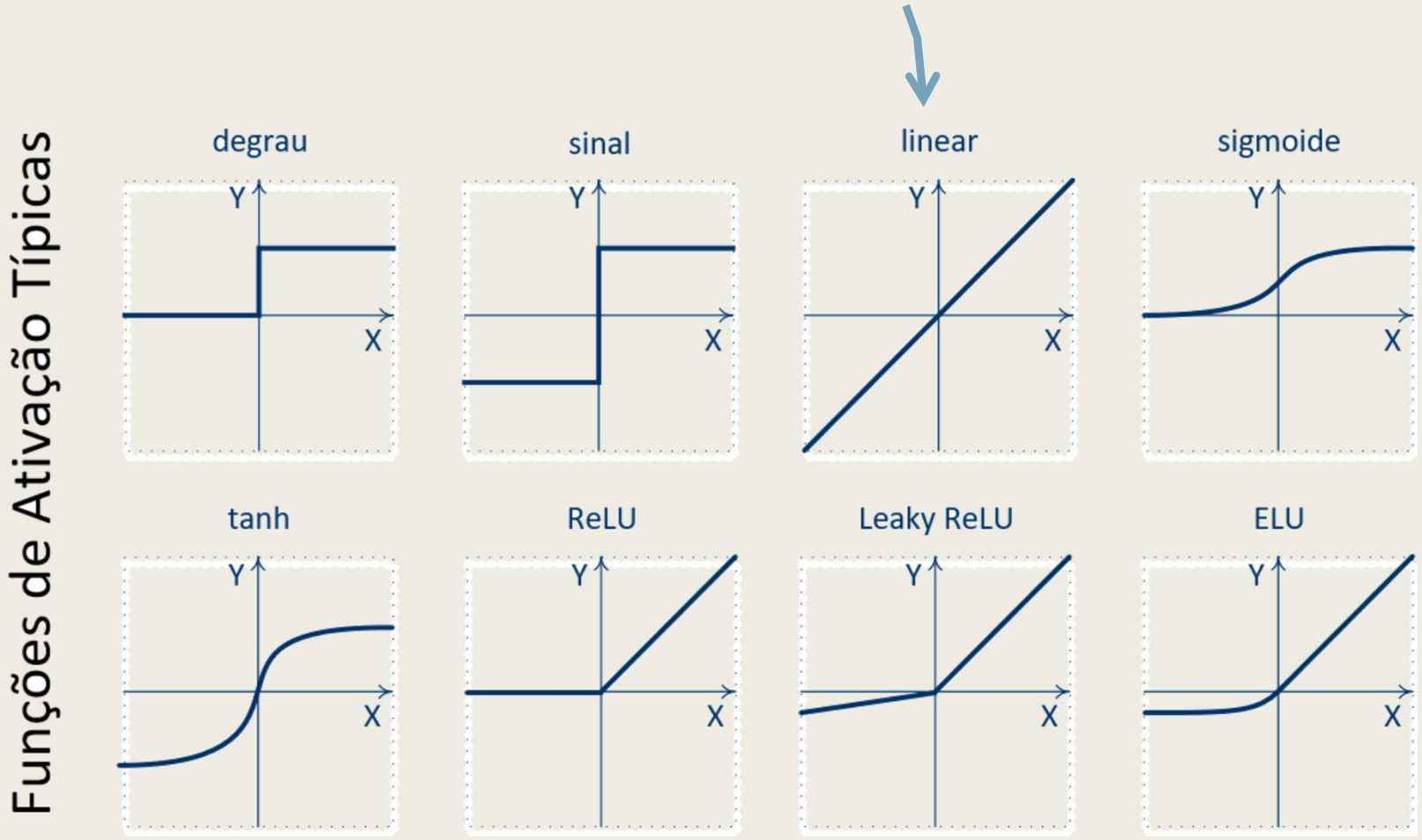
Activation Functions

- The output of the neuron will then depend on the activation function that is chosen:
  - *The sigmoid function, on the other hand, represents a continuous and differentiable approximation of the step function*



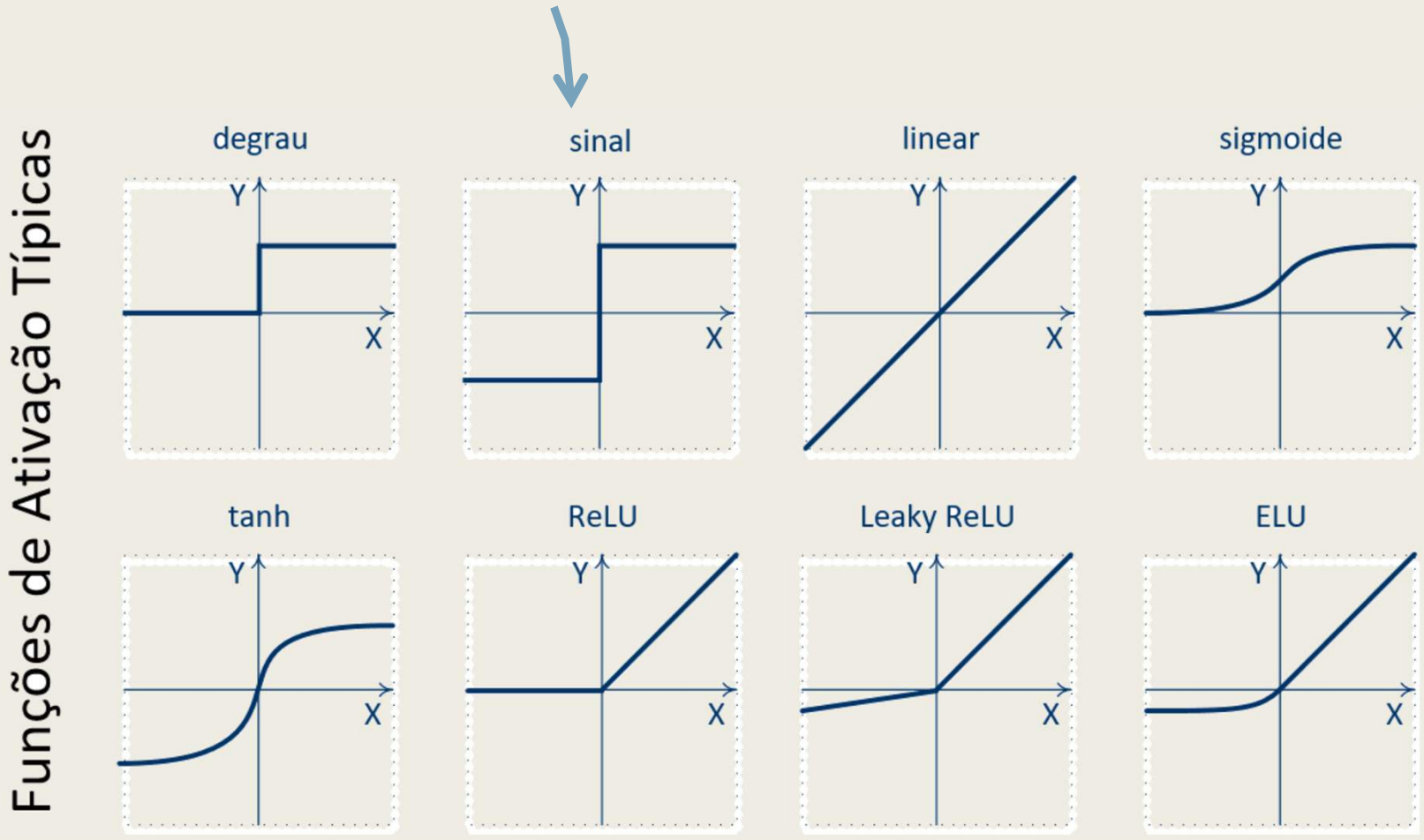
Activation Functions

- The output of the neuron will then depend on the activation function that is chosen:
  - *The use of the linear function will be limited to passing to the output a signal equal (or at least proportional) to the input*



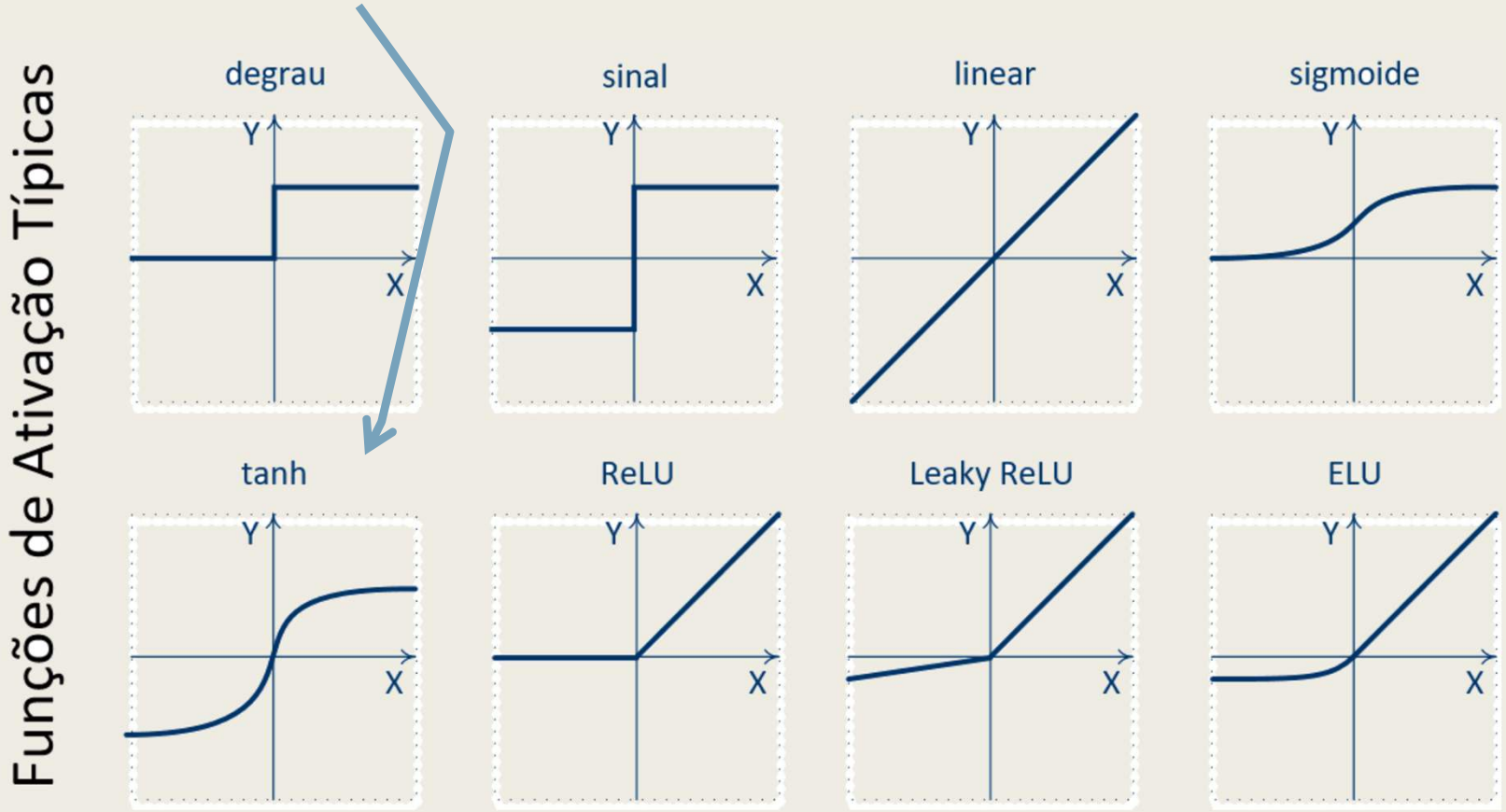
Activation Functions

- The output of the neuron will then depend on the activation function that is chosen:
  - *The signal function converts the input into −1 and +1 outputs in accordance with the negative or positive sign of the input*



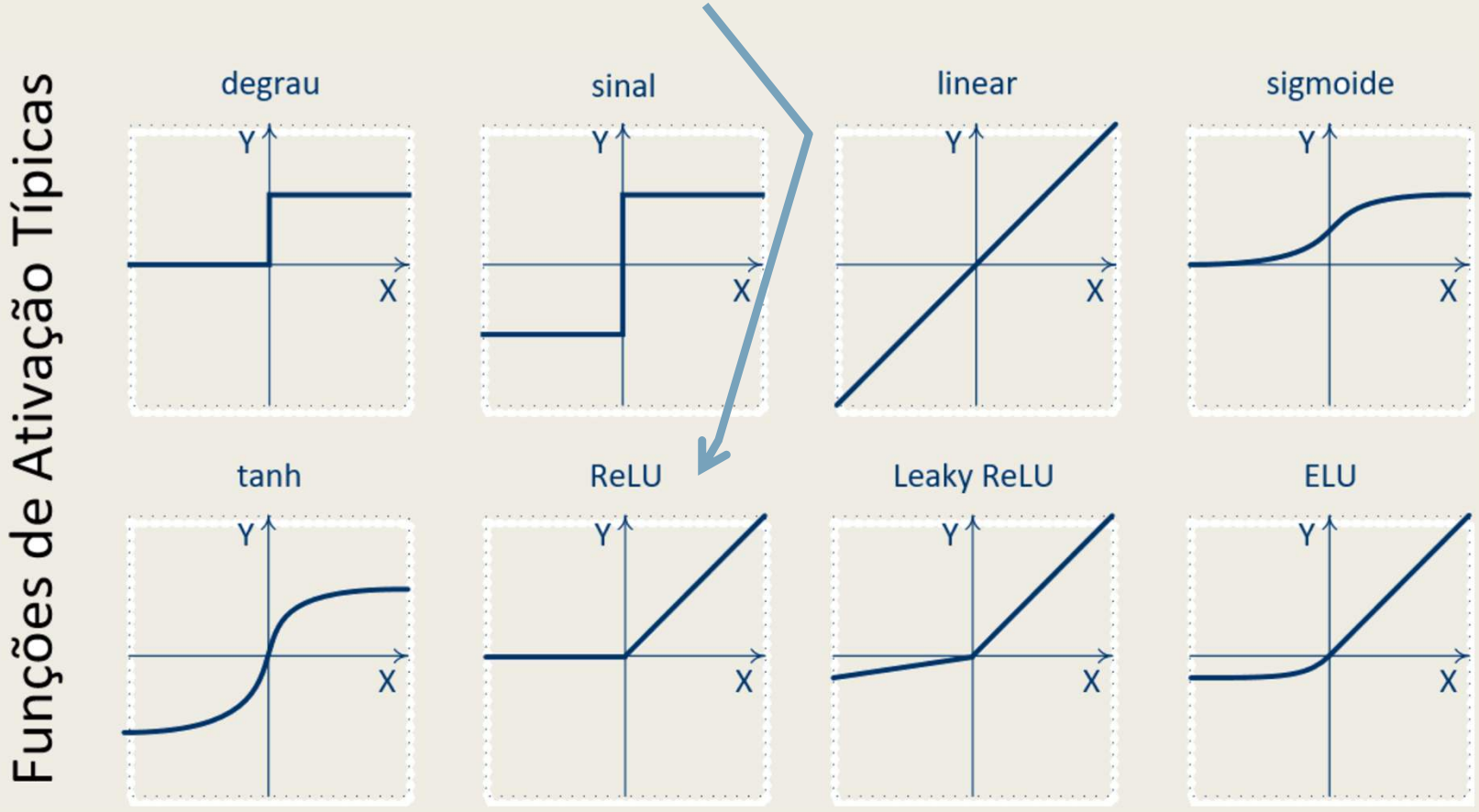
Activation Functions

- The output of the neuron will then depend on the activation function that is chosen:
  - *The tanh function (hyperbolic tangent) is a scaled version of the sigmoid, in order to make it symmetric around the origin (varies between -1 and 1)*



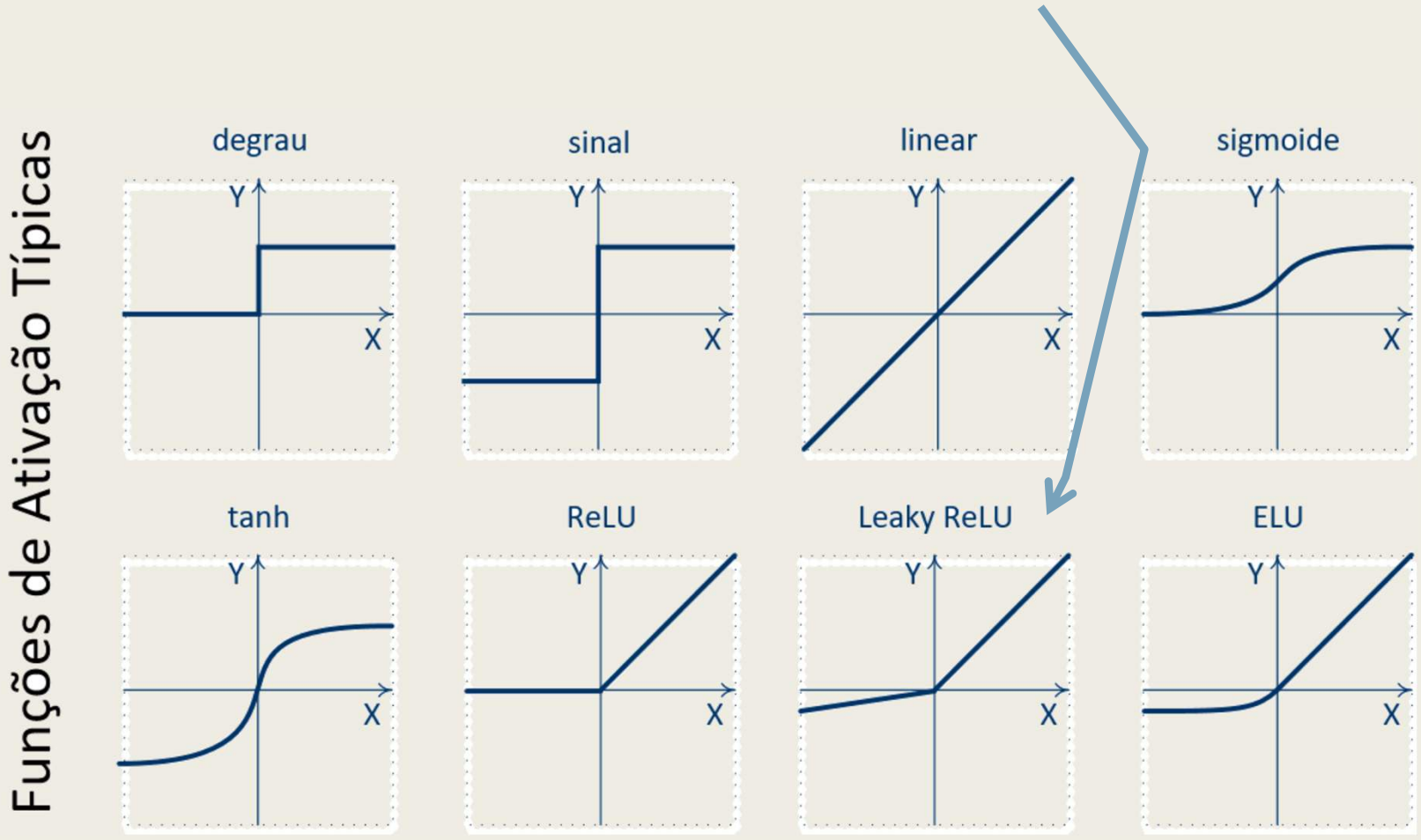
Activation Functions

- The output of the neuron will then depend on the activation function that is chosen:
  - *The ReLU function (rectified linear unit) allows only a few neurons to be activated at the same time, making the network sparse and therefore computationally more efficient*



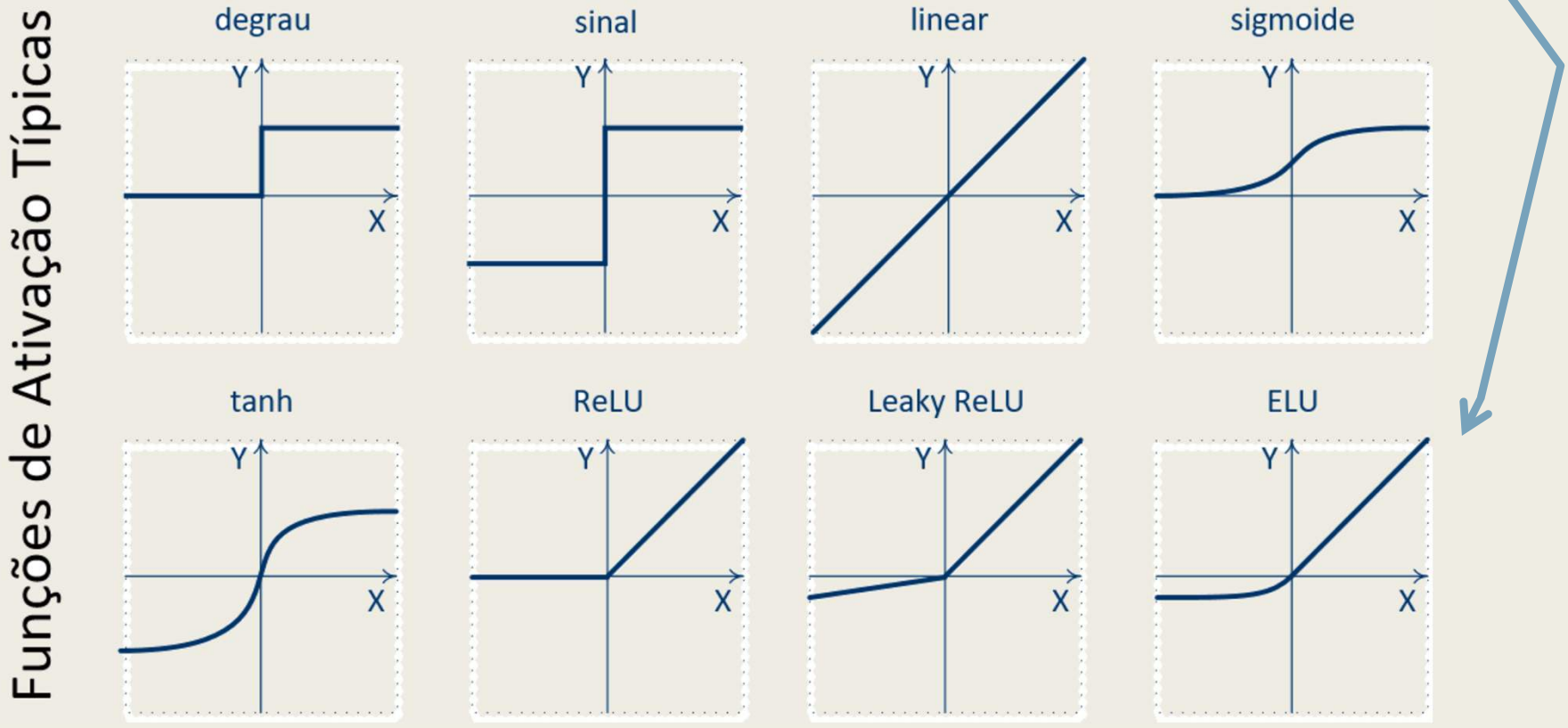
Activation Functions

- The output of the neuron will then depend on the activation function that is chosen:
  - *The Leaky ReLU function is an improved version of ReLU, which allows you to better handle gradients moving towards zero.*



Activation Functions

- The output of the neuron will then depend on the activation function that is chosen:
  - *The ELU function (exponential and linear unit) is another variant of the ReLU function, which replaces the 1st part of the function with an exponential curve, in order to increase the convergence and generalization capacity of the network*



Choice of Activation Function

There is no golden rule, and the choice will always depend on the problem to be addressed. However, some heuristics can be followed in this choice.

- The Step and Sign functions are rarely used because they are not differentiable, which makes it impossible to calculate the gradients, which are necessary to update the coefficients by backpropagation
- The Linear function is also rarely used, as the output of the network will be nothing more than a linear combination of the inputs, regardless of the layers that are used, thus serving only to solve problems of little complexity
- Sigmoid functions typically work best on classification problems
- The ReLU function, being of general application, is one of the most used today
- One rule that can be followed is to start by using the ReLU function and only then move on to other activation functions to see if the results improve

Perceptron

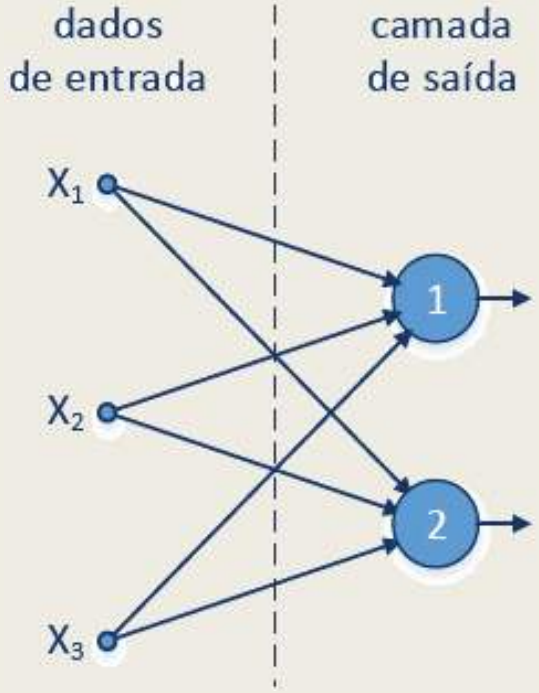
- A single neuron with its step-like activation function also makes up what is known as the Perceptron model,
  - *It is the simplest architecture of an artificial neural network*
  - *The Perceptron was the pioneering model developed in the 1950s and 1960s by Frank Rosenblatt, inspired by earlier work by Warren McCulloch and Walter Pitts.*
    - Given its simplicity, it is still useful today for the initial understanding of neural networks
  - *As we can easily understand, the Perceptron is a binary classifier,*
    - It can receive as inputs the explanatory variables of a problem and produce as output the respective classification: 0 or 1 (yes or no).

Neural Network Architecture

- Since the neurons are interconnected in a network, the **x** inputs of a neuron within the network will be no more than the outputs of the **d** neurons that precede it and are connected to it.
  - *On the other hand, if it is a neuron of the initial layer, the x inputs will correspond to the inputs of the problem itself*
  - *and since it is a neuron of the last layer, its output value will represent one of the values estimated by the network.*
- The various computational processing units, or artificial neurons, are arranged in one or more layers and interconnected by a large number of connections
  - *The number of layers, the number of neurons in each layer, the degree of connectivity, and the presence or absence of backpropagation connections define the architecture, or topology, of an NN*
    - There are several NN architectures, or topologies, with different characteristics. One of the most differentiating characteristics is whether the networks contain only forward connections (progressive or feedforward networks), or whether there are feedback loops, in which case they form backpropagation networks or recurrent networks.

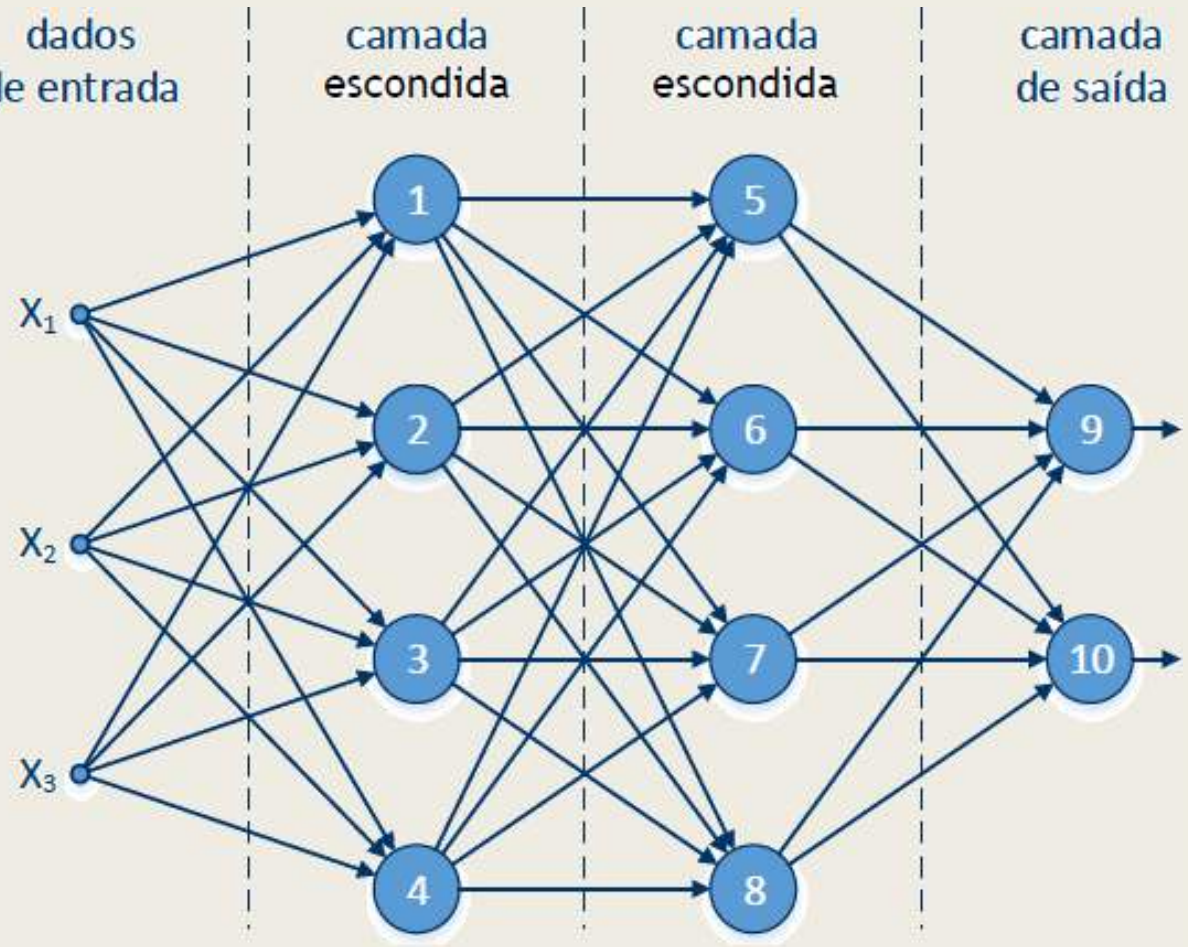
Single-layer Neural Networks

- In its simplest form, a single-layer progressive network is composed of:
  - *an input "layer", whose output values are fixed externally,*
  - *and by an output layer, containing all the neurons in the network.*
- They have the limitation of only being able to classify linearly separable objects (i.e., only if there is a hyperplan separating the data of the two classes)



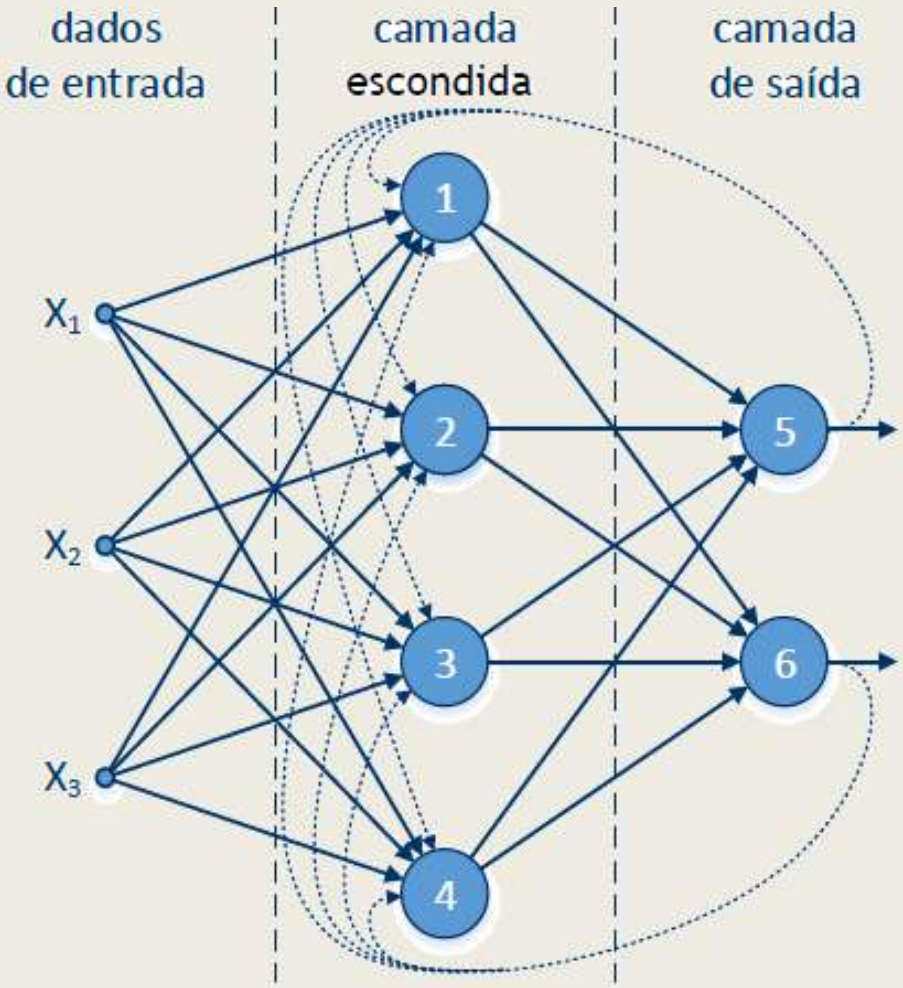
Multilayer Neural Networks

- They are distinguished by the fact that they have one or more intermediate layers (called hidden), whose neurons are also called intermediate neurons
  - *Their function is to do something usefully between the input and output of the network*
  - *They represent a very important sub-area of AI, which we will not develop, Deep Learning*
- By adding intermediate layers, the network's capacity to model more complex functions is increased.
- The most frequently used networks in this category are Multilayer Perceptron (MLP) networks



Recurrent Neural Networks

- They are fed back from the outputs to the inputs
  - *Due to this characteristic, they respond to X stimuli dynamically,*
    - That is, after applying a new input, the output is recalculated and used to modify the input again



Setting Up a Neural Network

- The first step for NNs to be able to induce a model from a dataset is to define its architecture
  - *The correct choice of activation function and network topology is decisive for successful learning or training*
    - For example, if the network is too small, it may be unable to solve the proposed problem.
    - On the other hand, if the network is too large, it may be incapable of generalizing (overfitting) and drastically increase processing time.
  - *The architecture of the network is, therefore, decisive in the processing and learning capacity of an NN.*
    - Generally, the most promising architecture is found through an exhaustive process of try and error, where different configurations are investigated, compared, and evaluated to select the one that has the best predictive capability.
      - *Although it is the most widely used approach, this blind search has the disadvantage of having a high processing time.*
      - *Alternatively, algorithms or tools can be used to perform a more efficient search*

Learning in a Neural Network

- The connections between the neurons of an NN have associated weights, which play, as already mentioned, an essential role in the acquisition of knowledge in a network.
- It is these weights that assume the appropriate values during the learning process, whose main objective is to obtain a set of desired and consistent outputs from a set of inputs. More specifically,
  - *After assigning an initial value to the weights of the connections, usually by a random process, all the observation arrays present in a training set are presented to the network, as a rule, more than once (several epochs)*
  - *Whenever one of these arrays is shown to the network, neurons in the first hidden layer receive these values and produce their outputs, passing them on to neurons in the next layer,*
    - and so on until we reach the last layer of neurons
  - *The value produced at the output of each of the last neurons is then compared with the result that is already known for the item in question*
    - The difference between the values verified at the output and the actual known values indicates the error made by the network for the item presented, and it is this error that, in some way, will be taken into account in the adjustment of the weights in a supervised learning
    - Throughout this process, the network weights gradually converge to certain values, and it is usually necessary to present the network with successive epochs of the training data until the input arrays produce the desired outputs.

Neural Network Training Algorithms

- Nowadays, there are several sets of training algorithms which use different learning rules
  - *One of the most popular for MLP networks is the backpropagation algorithm, which minimizes the squared error between the actual expected values and the values produced by the network*
    - The main idea behind this algorithm is that the error produced by the neurons in the hidden layers is estimated by successively propagating the errors made in the output layer.
    - It is an iterative algorithm that operates in two phases, first forward and then backwards,
      - *In the forward phase, an array of the training set is presented to the network, which calculates, with the current weights, the corresponding output and the associated error;*
      - *Then, in the backward phase, the error is successively backpropagated, calculating the gradient of the error made in each neuron in relation to the weights of its inputs,*
        - It is from the estimation of these intermediate errors that the appropriate fit for the net weights is found.

Summarizing...

An NN is defined by its architecture and learning algorithm

- *Therefore, in order to construct an NN, it is necessary to start by specifying the type and number of neurons and to particularize how they are interconnected, thus defining their topology.*
- *Then, initial values are assigned to the weights of the connections (usually random), and it starts with network learning, a phase in which each of the items in the training set is iteratively presented to the network*
  - For each of these training items, the weights are updated to minimise the error between the network prediction and the known result.
  - The process is repeated, successively showing the same training items to the network (several epochs), until the weights converge to appropriate values or until a limit number of iterations or epochs has been reached (the algorithm may not converge).

Learning Rate

- There is also an important parameter usually associated with the learning algorithm, called the **learning rate**, which allows us to speed up or slow down the training of the network
  - *This is usually a constant between 0 and 1, which the analyst can choose.*
  - *The higher this constant, the greater the change in weights in each iteration,*
    - thus increasing the speed of learning,
    - but also increase the possibility of oscillations occurring that make the algorithm unstable or non-convergent
  - *On the other hand, if the learning rate is too low,*
    - The possibility of the algorithm converging is in fact greater,
    - But, on the other hand, the network may take too long to learn.
  - *Another interesting option, which tries to mitigate the difficulties mentioned, is using a variable rate that decreases its value as the algorithm approaches the convergence value.*

Hyperparameters

- Among the most important parameters to fine-tune in a neural network in the search for the best ML model are:
  - *the number of hidden layers and the numbers of neurons in each of those layers;*
  - *The Rate of Learning;*
  - *the type of activation function to be used on neurons*
  - *and the maximum number of iterations or epochs used to train the network.*

Solve exercise #23

from the book of exercises