NEURAL NETWORKS

Análise de Dados em Informática
Licenciatura em Engenharia Informática
ISEP/IPP

Ana Maria Madureira
2022/2023

Fontes:

- Tom Mitchell, Machine Learning. McGraw-Hill, 1997.
- Catarina Silva e Bernardete Ribeiro, Aprendizagem Computacional em Engenharia, Imprensa da Universidade de Coimbra, 2018
- Chantal D. La and Daniel T. Larose, Data Science Using Python and R, John Wiley & Sons, Inc., 2019

DT, NN and KNN

Decision Trees are among the most used and have been applied to a large number of fields (since medial diagnosis to credit risk management in banking

Neural networks are gaining popularity in the computer vision and pattern recognition field.

K-nearest neighbors models are still commonly and successfully being used is:

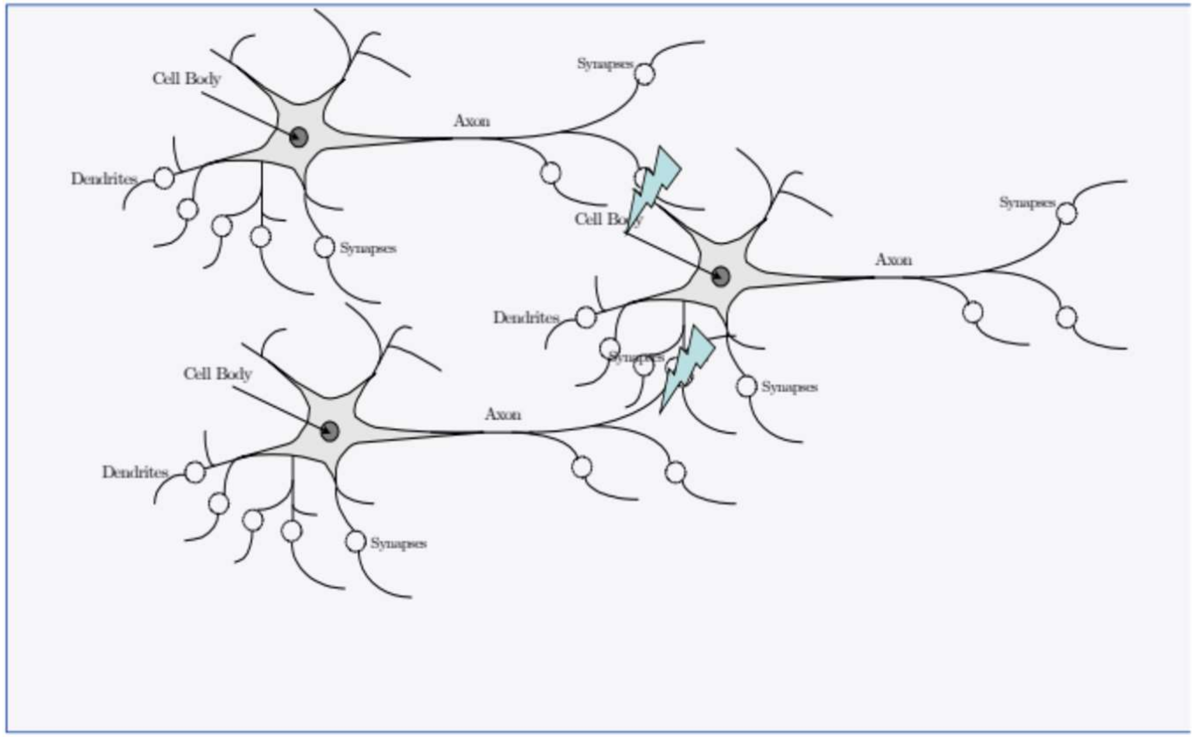
- in the intersection between computer vision, pattern classification, and biometrics(to make predictions based on extracted geometrical features).
- recommender systems (via collaborative filtering and outlier detection

Neural Network - Historic Perspective

1943 - McCulloch and Pitts proposed the model of an artificial neuron, also called the McCulloch-Pitts model
1948 - Weiner published the book “Cybernetics”, in 1961, where aspects of learning and self-organization were discussed
1949 - Hebb published the book “The Organization of Behavior”, in which Hebb's Rule was proposed
1958 - Rosenblatt introduced the simple one-layer network - Perceptron
1969 - Minsky and Paper published the book “Perceptrons” in which they demonstrated the limitations of a single layer network. The area then had a period of stagnation
1982 - Hopfield published several works dedicated to the Hopfield Recurring Networks
1982 - Kohonen developed the self-organizing maps with his name
1986 - The backpropagation algorithm for multi-layer networks was rediscovered, since then there has been a notable growth in the area
1985 - Boltzmann machines were proposed by Ackley, Hinton and Sejnowski
1989 - Yann LeCun proposed convolutional neural networks (CNN), initially advocated and developed by Fukushima in 1980
1992 - Neal moves forward with Bayesian networks
1995 - Recurring long memory networks were presented by Jurgen Schmidhuber
2006 - The appearance of Deep Neural Networks launched a new interest in the area with numerous applications
2014 - Adverse networks appeared by GoodFellow

NEURAL NETWORKS

NN models are inspired by biological neuron structures and computations.



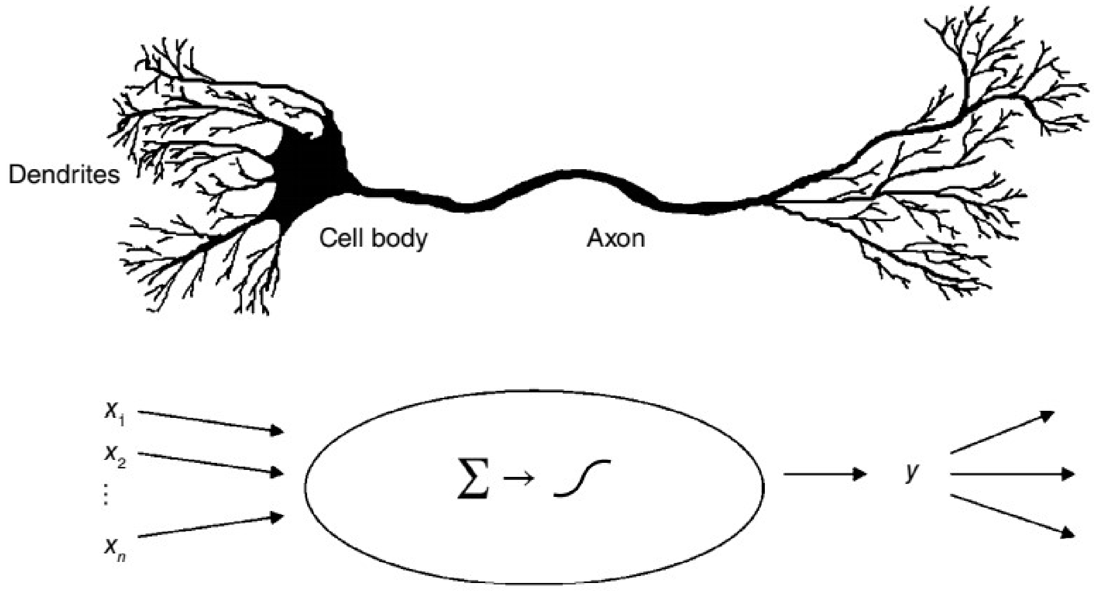
Real neuron versus artificial neuron model

NEURAL NETWORKS

- Artificial neural networks have a surprising number of characteristics observed in the human cognitive process: learning by experience; generalization
from examples and abstraction of the essential characteristics of information that contains irrelevant facts.
- Learning can be defined as the ability to perform new tasks that could not have been done before, or to improve previous performance as a result of changes produced by the learning process.
- Artificial neural networks can modify their behavior in response to stimuli produced by the environment: regulating the intensity of the connection between the processing units; adapting the synapse weights; recognizing the information presented to their neurons.
- To build a neuronal network it is necessary to consider the following mains steps:
    - Define the number and type of neurons;
    - Define how they connect (topology);
    - Start the connection weights;
    - Learn which weights are appropriate for the problem (learning)

NEURAL NETWORKS

Neural networks represent an attempt at a very basic level to imitate the type of nonlinear learning that occurs in the networks of neurons found in nature, such as the human brain

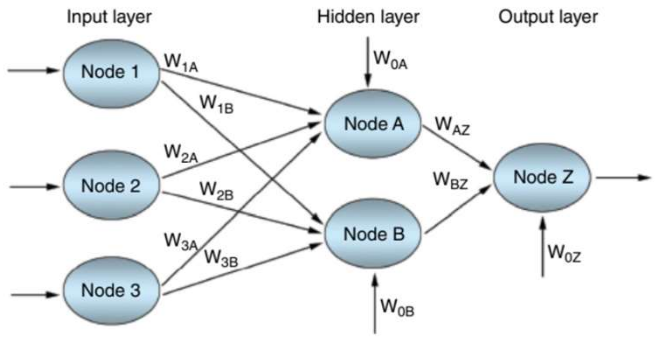


Real neuron versus artificial neuron model

NEURAL NETWORKS

A neural network consists of a layered, feedforward, completely connected network of artificial neurons or nodes:

- The feedforward nature of the network restricts the network to a single direction of flow and does not allow looping or cycling.
- Most networks consist of three layers: an input layer, a hidden layer, and an output layer.
  - There may be more than one hidden layer, although most networks contain only one, which is sufficient for most purposes.
- The neural network is completely connected, meaning that every node in a given layer is connected to every node in adjoining layers, although not to other nodes in the same layer.
  - Each connection between nodes has a weight (e.g. W1A) associated with it.
  - At initialization, these weights are randomly assigned to values between 0 and 1.



Simple example of a neural network

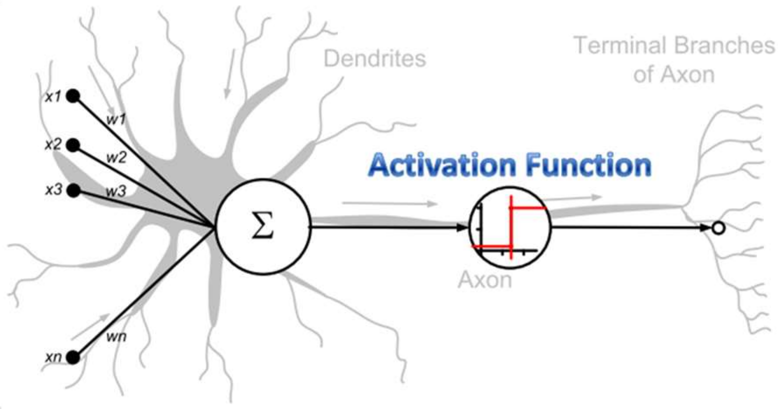
- The number of hidden layers and the number of nodes in each hidden layer are both configurable by the analyst. How many nodes should one have in the completely connected network of artificial neurons or nodes.
- The feedforward nature of the network restricts the network to a single direction of flow and does not allow looping or cycling.

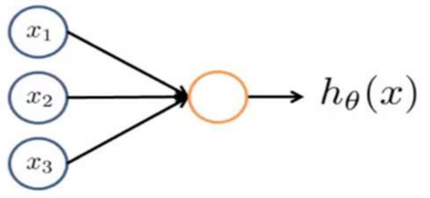
How many nodes should one have in the hidden layer?

- Having more nodes in the hidden layer increases the power and flexibility of the network for identifying complex patterns – so, might be tempted to have a large number of nodes in the hidden layer.
  - an overly large hidden layer leads to overfitting, memorizing the training set at the expense of generalizability to the validation set
  - if overfitting is occurring, may consider to reduce the number of nodes in the hidden layer.
  - if the training accuracy is unacceptably low, may consider increasing the number of nodes in the hidden layer.

NEURAL NETWORKS – Neuron Model

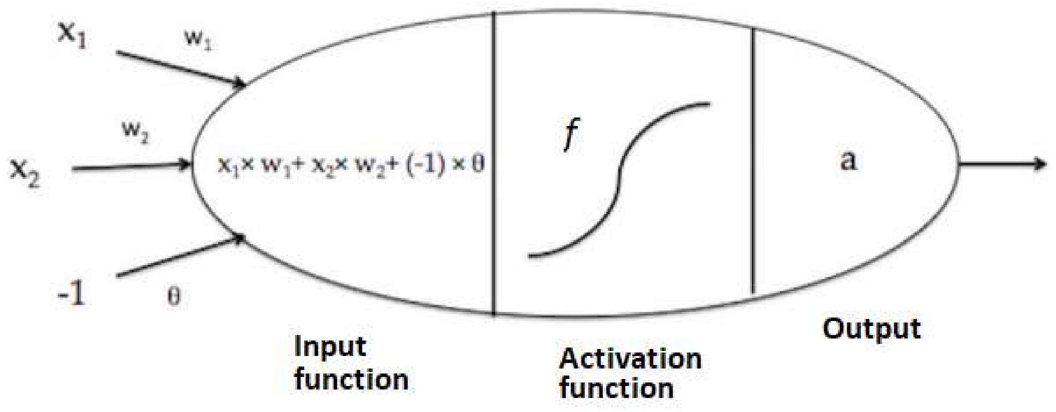
- An artificial neuron (similar to the biological) has the function of performing a simple operation that consists of receiving signals from the input connections and calculate a new output value that is and sent out. Two phases can be identified:
    - Calculate the weighted sum (combination function) of the input values;
    - Calculate the neuron activation value using a function - the activation function or transfer function, which has an input argument - the value calculated in the previous step.





Model of artificial neuron (unit, code)

Artificial Neuron - mathematical model



f : activation function (or transfer function)
∑: input function or combination function
w : input weight
θ: bias or offset
x_1, x_2: inputs

- 2 inputs x_1, x_2 are visible with 2 associated weights w_1, w_2 and one input special value -1, whose weight is called offset or bias and whose weight it is usually represented by the letter b or the Greek letter θ.
- The combination function is the weighted sum of these inputs is carried out in the first step of the operation of the neuron resulting in $\sum = x_1 \times w_1 + x_2 \times w_2 + (-1) \times \theta$.
- the activation function *f*, has as objective to determine whether the previously calculated weighted sum is sufficient to activate the neuron. the most common functions are the step function (sometimes also referred to as a signal function when the lower value is -1 instead of 0), the linear function and the sigmoid function.

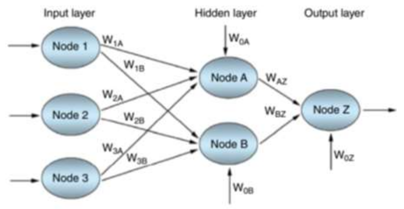
Combination Function

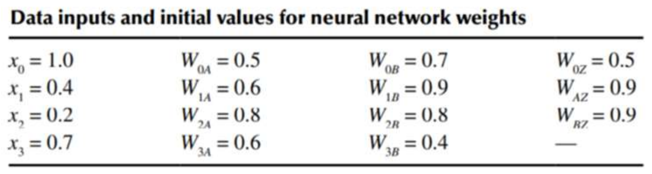
- The nodes in the hidden layer and the output layer collect the inputs from the previous layer and combine them using a combination function
- The combination function *Σ* produces a linear combination of the node inputs and the connection weights into a single scalar value, which we will term net. Thus, for a given node j, where x_{ij} represents the i^{th} input to node *j*, w_{ij} represents the weight associated with the i^{th} input to node *j*, and there are *I+1* inputs to node j. Note that x_1 , x_2 , …, x_I represent inputs from upstream nodes, while x_0 represents a constant input, analogous to the constant factor in regression models, which by convention uniquely takes the value $x_{0j} = 1$. Thus, each hidden layer or output layer node *j* contains an "extra" input equal to a particular weight $W_{0j}\, x_{0j} = W_{0j}$, such as W_{0B} for node B

$$net_j = \sum_i W_{ij} x_{ij} = W_{0j} x_{0j} + W_{1j} x_{1j} + \cdots + W_{Ij} x_{Ij}$$

Combination Function - Example

- Details of neural network, showing input to node A, combination function, and output from node A.
- we see that node A has combined its inputs into a net input of 1.32. For node A, this combination function netA = 1.32 is then used as an input to an activation function. In biological neurons, signals are sent between neurons when the combination of inputs to a particular neuron crosses a certain threshold and the neuron “fires.” This is nonlinear behavior, since the firing response is not necessarily linearly related to the increment in input stimulation.

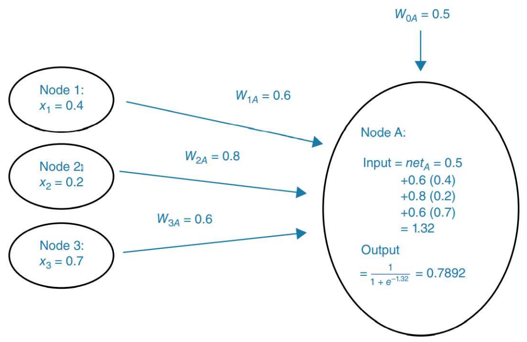




Data inputs and initial values for neural network weights

$x_0 = 1.0$	$W_{0A} = 0.5$	$W_{0B} = 0.7$	$W_{0Z} = 0.5$
$x_1 = 0.4$	$W_{1A} = 0.6$	$W_{1B} = 0.9$	$W_{AZ} = 0.9$
$x_2 = 0.2$	$W_{2A} = 0.8$	$W_{2B} = 0.8$	$W_{BZ} = 0.9$
$x_3 = 0.7$	$W_{3A} = 0.6$	$W_{3B} = 0.4$	—

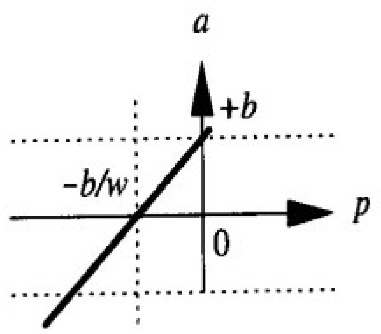
$$net_A = \sum_i W_{iA}x_{iA} = W_{0A}(1) + W_{1A}x_{1A} + W_{2A}x_{2A} + W_{3A}x_{3A}$$
$$= 0.5 + 0.6(0.4) + 0.80(0.2) + 0.6(0.7) = 1.32$$



Details of neural network, showing input to node A, combination function, and output from node A.

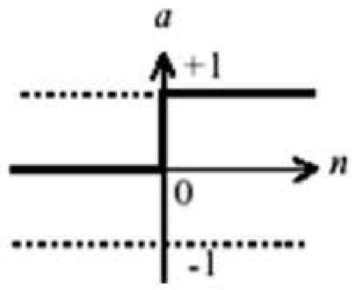
Activation functions

Linear



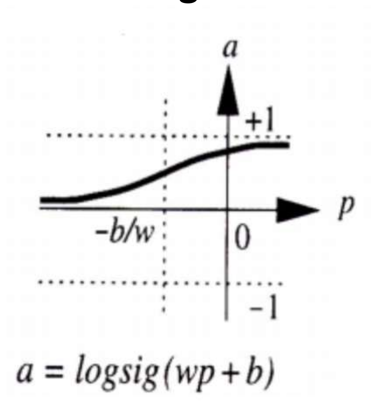
The bias b makes horizontal displacements of the linear function

Step



The bias b makes horizontal displacements of the step function

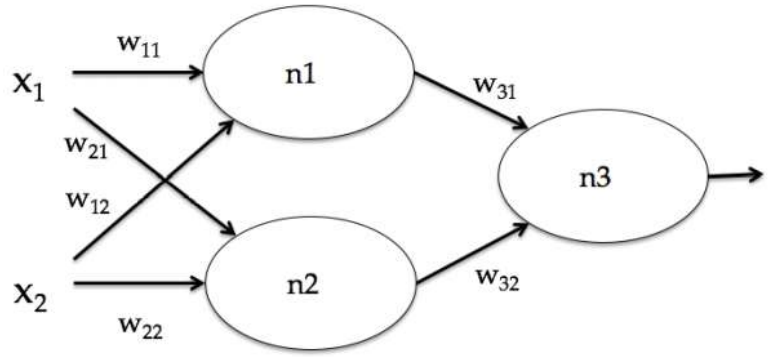
Sigmoid



The bias b makes horizontal displacements of the sigmoid function

Feed-forward network

- In general, a neuron, even with several inputs, is not enough to solve the majority of real problems. Several neurons are required to work in parallel, in what is usually called a layer of neurons, or even several layers of neurons constituting a multilayered architecture
- A network with only forward or direct connections, with three layers of neurons, with the input layer ([x_1 x_2]). Each layer has its weight matrix *w*, the bias is not represented at this case.
- The output layer has only one neuron (n_3) and the intermediate layer, called a hidden layer, it consists of two neurons (n_1 and n_2), the various layers can have a different number of neurons.
- When input layer and the output layer are all called the hidden layer (1st hidden layer, 2nd hidden layer, and so on from the input to the output). The outputs of the input layer constitute the inputs of the hidden layer, and so on, so that each layer can be interpreted as an isolated layer.
- Faced with a problem to be solved with feed-forward neural networks, the number of neurons in the input and output layers are usually easy to define by the problem definition itself.
- The number of neurons in each hidden layer and the number of hidden layers is more subjective. There are many heuristics to choose the structure of a neural network, there is no method that allows to choose exactly the best structure for a neural network.
- If the network is too small, you may not be able to solve the proposed problem. While if the network is too large, despite being able to memorize the examples presented, it may prove unable to generalize beyond those examples - it may lose its most important capacity - the ability to learn (overfitting)



Model of a neuronal network with three layers of neurons

Neural Network Topologies

- There are several topologies of artificial neural networks with specific characteristics:
  - networks have only forward connections (feed-forward networks)
  - if there are feedback loops (recurrent networks).
  - have a supervisor or teacher, that is, if they are supervised or if they have autonomous learning algorithms(self-organized).

Multilayer NNs

- Multilayer NNs are more powerful than monolayer networks, having more applications. For example, unlike a monolayer network, a network with three layers, with sigmoid activation function in the hidden layer and linear activation function in the output layer, it is able to approximate any function with a small error, provided that properly trained.

The binary perceptron: training and learning

- Learning rule or training algorithm: - a systematic procedure to modify the weights and the bias of a NN such that it will work as we need
- The most common learning approaches:
  - supervised learning - A set of Q examples of correct behavior of the NN is given (inputs p, outputs t)
  - reinforcement learning - receives only a classification that stimulate good performances.
  - unsupervised learning - has only inputs, not outputs, and learns to categorize them (dividing the inputs in classes, as in clustering)

The binary perceptron:

- The perceptron learning rule is a supervised one
- It is simple, but powerful
- With one single layer, the binary perceptron can solve only linearly separable problems.
- The problems non linearly separable can be solved by multilayer architectures.

Backpropagation

- **How does the neural network learn?**
- As each observation from the training set is processed through the network, an output value is produced from the output node.
- This output value is then compared to the actual value of the target variable for this training set observation and the error (actual – output) is calculated. This prediction error is analogous to the prediction error in linear regression models. To measure how well the output predictions fit the actual target values, most neural network models use the sum of squared errors (SSE):

$$\text{SSE} = \sum_{\text{records}} \sum_{\text{output nodes}} (\text{actual} - \text{output})^2$$

- where the squared prediction errors are summed over all the output nodes and over all the records in the training set.
- The problem is to construct a set of model weights that will minimize the SSE. In this way, the weights are analogous to the parameters of a regression model. The “true” values for the weights that will minimize SSE are unknown and our task is to estimate them, given the data.

Backpropagation – Algorihtm

- The backpropagation algorithm does the following:
  - It takes the prediction error (actual – output) for a particular record and percolates the error back through the network.
  - Assign partitioned responsibility for the error to the various connections.
  - The weights on these connections are then adjusted to decrease the error, using gradient descent.

Neural Networks - conclusions

- The main benefit of neural networks is that they are quite robust for noisy, complicated, or nonlinear data, due to the nonlinear nature of the activation function.
- The main drawback of neural networks is that they are relatively opaque to human interpretation

Atividade: Neural Networks

- As redes neuronais representam uma tentativa de imitar o quê?
- Qual é o principal benefício das redes neuronais para modelação?
- Descreva a principal desvantagem de uma rede neuronal.