DRIVES

Development and Research on Innovative Vocational Education Skills

ARTIFICIAL INTELLIGENCE OVERVIEW

U1.E3. UNDERPINNINGS OF DEEP LEARNING

Computer Vision Expert

May 2021, Version 1

Co-funded by the Erasmus+ Programme of the European Union

The Development and Research on Innovative Vocational Educational Skills project (DRIVES) is co-funded by the Erasmus+ Programme of the European Union under the agreement 591988-EPP-1-2017-1-CZ-EPPKA2-SSA-B. The European Commission support for the production of this publication does not constitute endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

LEARNING OBJECTIVES

The student is able to

CVE.U1.E3.PC1	Define deep learning.
CVE.U1.E3.PC2	Understand the connection and the differences between artificial intelligence, machine learning and deep learning.
CVE.U1.E3.PC3	Understand the nature, purpose and applications of an artificial neuron.
CVE.U1.E3.PC4	Recognize and understand the similarities between a computational neuron and a human neuron.

LEARNING OBJECTIVES

The student is able to

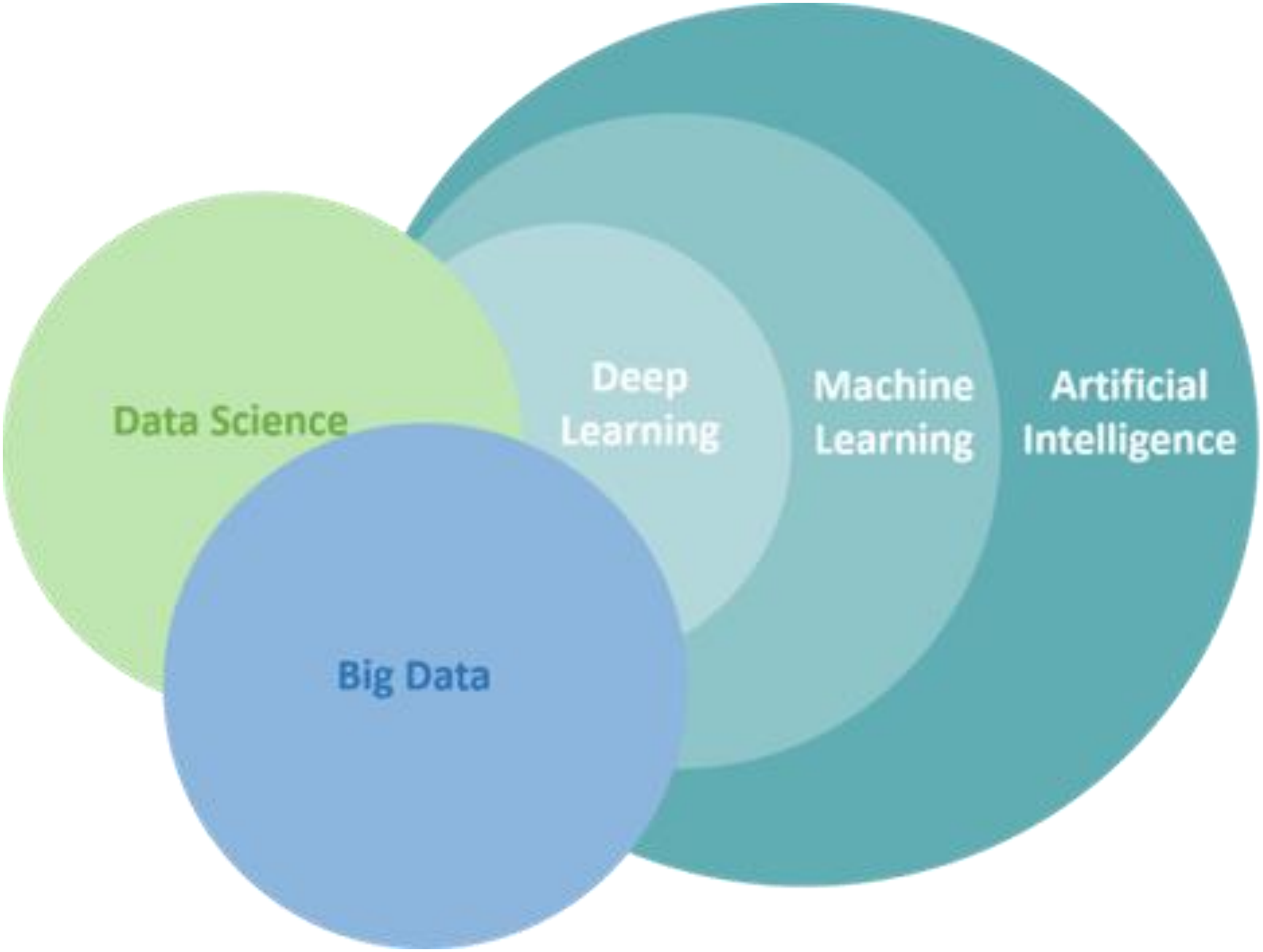
CVE.U1.E3.PC5	Know the most commonly used activation functions for artificial neurons.
CVE.U1.E3.PC6	Define and explain what an artificial neural network is.
CVE.U1.E3.PC7	Know some of the most common neural networks like perceptron, multi-layer perceptron, feed-forward, backpropagation, etc.
CVE.U1.E3.PC8	Define Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) and explain their differences.

DRIVES
Development and Research on Innovative Vocational Education Skills

Conceptual Definition: Computer program that is capable of identify what something is....

Technical Definition: Deep Learning consists in using a neural networks with several layers of nodes between input and output

Definition: Subfield of machine learning based on algorithms inspired in artificial neural networks with many layers of nodes, that are capable of identify what something is....


Data Science
Deep
Learning
Machine
Learning
Artificial
Intelligence
Big Data

ARTIFICIAL INTELIGENCE VS MACHINE LEARNING

ARTIFICIAL INTELIGENCE	MACHINE LEARNING
AI is defined as acquisition of knowledge intelligence	ML is defined as the acquisition of knowledge or skill
The aim is to increase the chance of success	The aim is to increase accuracy
It works as a computer program that does smart work	It is a simple machine that takes data and learns from it
The goal is to simulate natural intelligence to solve complex problems	The goal is to learn from data a specific task to maximize machine performance on this task

ARTIFICIAL INTELIGENCE VS MACHINE LEARNING

ARTIFICIAL INTELIGENCE	MACHINE LEARNING
AI is decision making	ML enables systems to learn new things from data
It leads to the development of a system that mimics human behavior to respond to circumstances	It is involved in the creation of self-learning algorithms
AI pursues the optimal solution	ML will only come up with a solution that is either optimal or not
Leads to intelligence or wisdom	Leads to knowledge

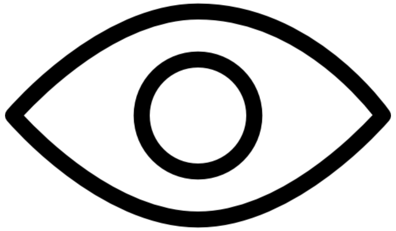
DEEP LEARNING VS NEURAL NETWORK

Deep Learning	Neural Network
Referring to the depth of layers in a neural network	It is a component of deep learning process
It is a subset of machine learning	
The architecture of a Deep Learning model includes: Unsupervised Pre-trained Networks , Convolutional Neural Networks, Recurrent Neural Networks, Recursive Neural Networks	The architecture of a Neural Network includes: Feed Forward Neural Networks, Recurrent Neural Networks, Symmetrically Connected Neural Networks
Leads with the transformation and extraction of feature which attempts to establish a relationship between stimuli and associated neural responses present in the brain	Leads with the transition data in the form of input values and output values through connections

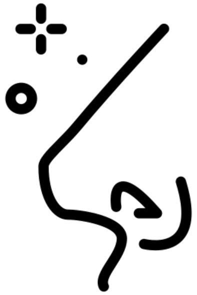
NEURON

Neurons or nerve cells, send and receive signals from your brain.

The nervous system controls all our senses:

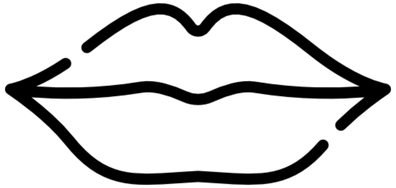


Vision



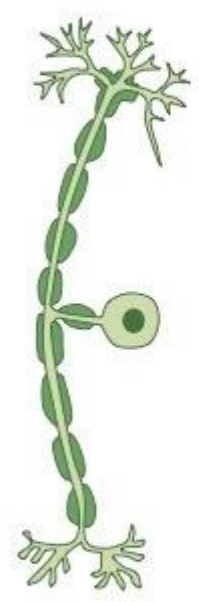
Smell

Hearing



Taste

In short, neurons are part of a system that deals with all the sensations that control our daily lives.



Sensory Neuron

They receive stimulus, which can come from the organism itself or the environment.

Motor Neurons

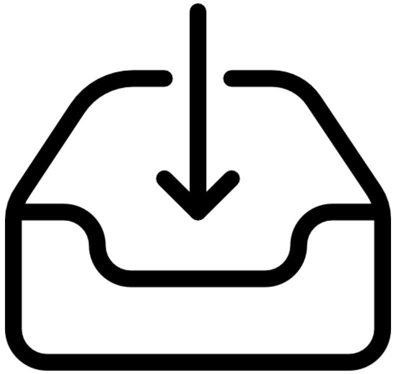
They are responsible for conducting nerve impulses to effector organs, such as muscles and glands.

Interneurons

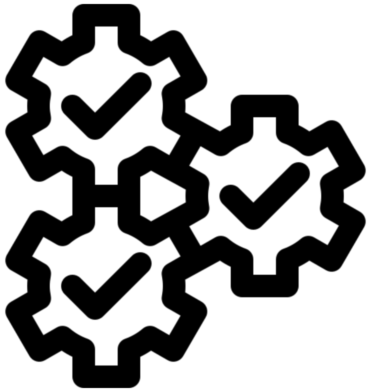
They guarantee the connection between neurons.

FUNCTIONS OF A NEURON

There are roles, one for the each classe of neurons. All of them have three basic functions:



1. Receive signals (or information).

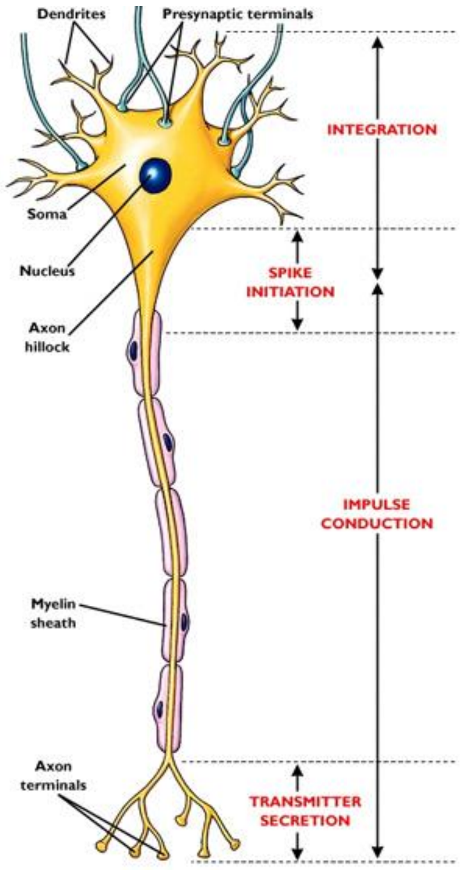


2. Integrate input signals (to determine whether this information should be passed on or not).

3. Communicate signals to target cells (other neurons or muscles or glands).

These neuronal functions are reflected in the anatomy of the neuron.

Dendrite	They are neuron extensions. They guarantee the reception of stimuli, leading the nervous impulse towards the cellular body.
Axon	Extension that guarantees the conduction of the nervous impulse. Each neuron has only one axon. Around the axon is an electrical insulation called myelin sheath. The sites where this sheath fails are called Ranvier's nodules.
Cellular Body	Place where the core is. Most of the cellular organelles are also located in the cellular body. In addition, it is from where the extensions of this cell originate.



The neuron collects signals from Dendrites, and the Soma cells sums up all the signals collected, and when the summation reaches the threshold the signal pass through the axon to the other neurons

The artificial neuron is a **simplified and simulated model** of the real neuron as well as its basic characteristics.

These characteristics are the adaptation and the representation of knowledge based on connections.

In 1943 MacCulloch & Pitts developed the first mathematical model of a neuron = node.

After some time, they realized that the combination of several neurons produces high computational power.

They associated the response of all or nothing (characteristic of a neuron) and realized that they only perform logical functions.

Concluding, they realized that a set of neurons can be represented by an artificial neuronal network

WHY ARTIFICIAL NEURONS?

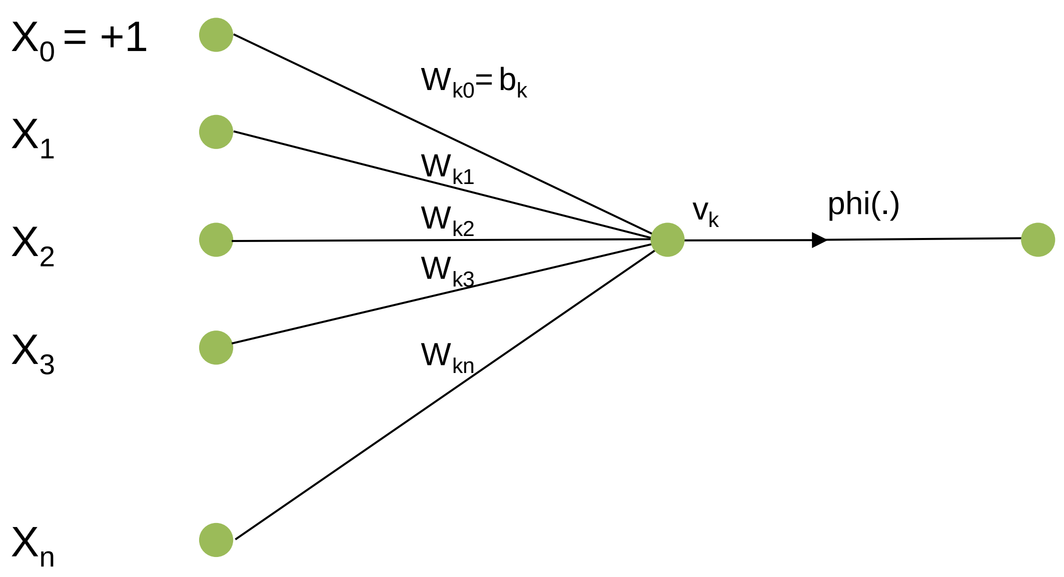
Neural networks, which consist of artificial neurons, have an excellent behavior for helping people with complex day-to-day problems

They are able to:

- Learn and model the relationships between inputs and outputs that are nonlinear and complex
- Model highly volatile data and variances needed to predict rare events.
- Reveal hidden relationships, patterns and predictions

For a given artificial neuron k:

- m + 1 inputs
- signals x_0 through x_m
- weights w_{k0} through w_{km}
- *Transfer function phi*

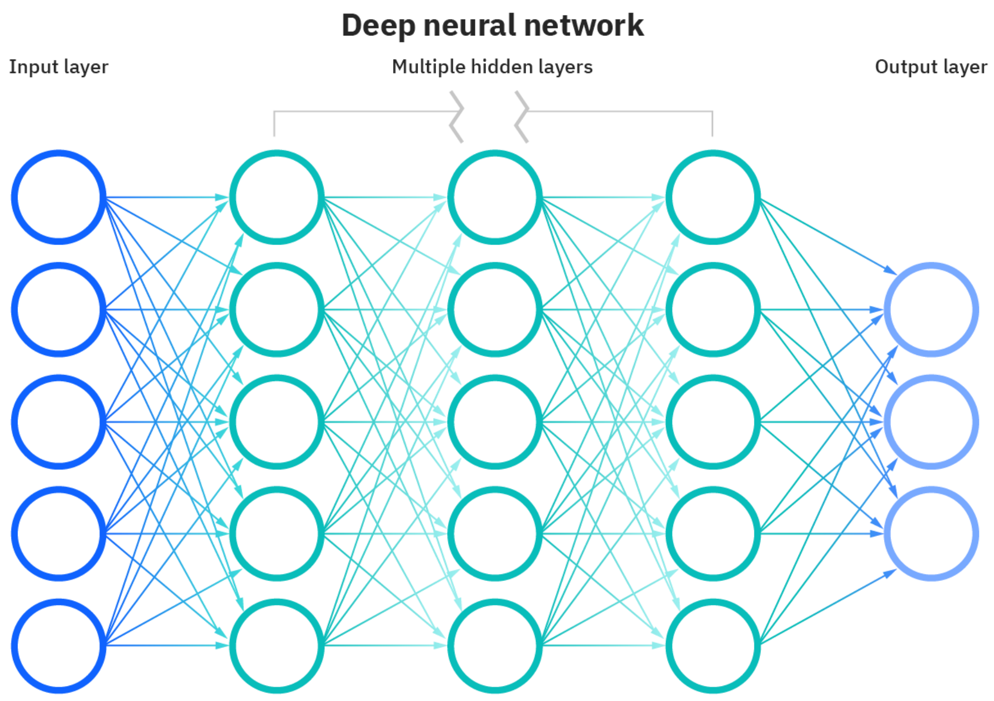


In the Neural Network, the neurons are arranged into multiple layers.

Input Layer : This layer accepts input features. It provides information from the outside world to the network, no computation is performed at this layer, nodes here just pass on the information(features) to the hidden layer.

Hidden Layer : Nodes of this layer are not exposed to the outer world, they are the part of the abstraction provided by any neural network. Hidden layer performs all sort of computation on the features entered through the input layer and transfer the result to the output layer.

Output Layer : This layer bring up the information learned by the network to the outer world.



ARTIFICIAL NEURON APPLICATIONS

- Credit card fraud detection.
- Optimization of logistics for transportation networks.
- Character and voice recognition (Natural language processing).
- Medical and disease diagnosis.
- Targeted marketing.
- Financial predictions for stock prices, currency, options, futures, bankruptcy and bond ratings.
- Robotic control systems.
- Electrical load and energy demand forecasting.
- Process and quality control.
- Chemical compound identification.
- Ecosystem evaluation.
- Computer vision to interpret raw photos and videos (Medical Imaging, Robotics and Facial Recognition).

SIMILLARITIES BETWEEN ARTIFICIAL NEURON AND HUMAN NEURON

DRIVES
Development and Research on Innovative Vocational Education Skills

Both use electrical signals to send messages.

Both can do math and other logical tasks.

Both have a memory that can grow.

Both can change and be modified.

Both can adapt and learn.

Both transmit information.

Both need energy.

MAIN DIFFERENCES BETWEEN ARTIFICIAN NEURON AND HUMAN NEURON

Size

The human brain contains about 86 billion neurons and more than a 100 trillion synapses (connections). The number of “neurons” in artificial networks is much less than that.

Speed

Biological neurons can fire about 200 times a second on average. Signals travel at different speeds depending on the type of the nerve impulse, ranging from 0.61 m/s up to 119 m/s.

Power consumption

The human brain consumes about 20% of all the human body’s energy. Our machines are way less efficient than biological systems.

Topology

All artificial layers compute one by one, instead of being part of a network that has nodes computing asynchronously.

Fault-tolerance

Biological neuron networks due to their topology are also fault-tolerant. Information is stored redundantly so minor failures will not result in memory loss. They don’t have one “central” part.

Learning

We still do not understand how brains learn, or how redundant connections store and recall information

ACTIVATION FUNCTIONS

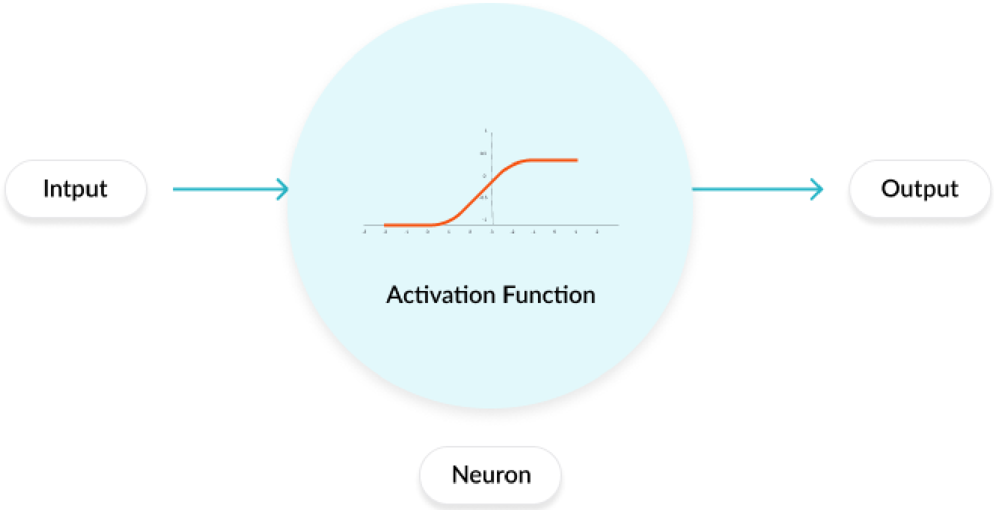
Activation functions are mathematical equations that determine the output of a neural network.

The function is attached to each neuron in the network, and determines whether it should be activated (“fired”) or not.

Activation functions also help normalize the output of each neuron to a range between 1 and 0 or between -1 and 1.

The Activation Functions can be basically divided into 2 types:

1. **Linear Activation Functions**
2. **Non-linear Activation Functions**



SIGMOID FUNCTION

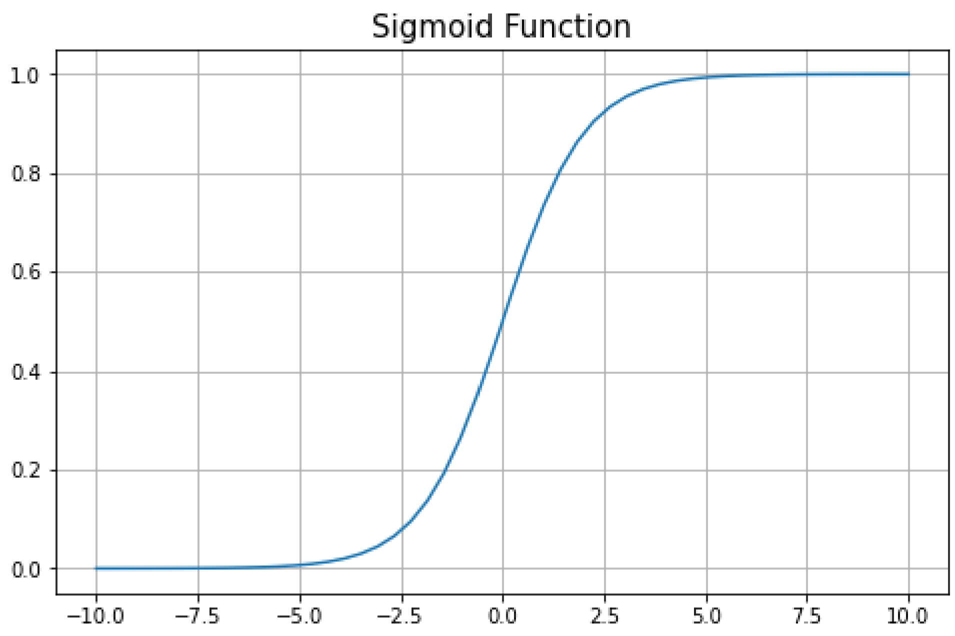
It is very used because it exists between 0 and 1. Also because of that, it is the most used for models to predict probability.

$$Sigmoid\ Function = \frac{1}{1 + e^{-x}}$$

It is **Differentiable** and **Monotonic**

Change in y-axis w.r.t. change in x-axis.It is also known as slope.

A function which is either entirely non-increasing or non-decreasing.



Downside: The derivate tends towards zero as we move away from zero. The “learning” process of a neural network depends on the derivative because the weights are updated based on the gradient which basically is the derivate of a function.

TANH (HYPERBOLIC TANGENT)

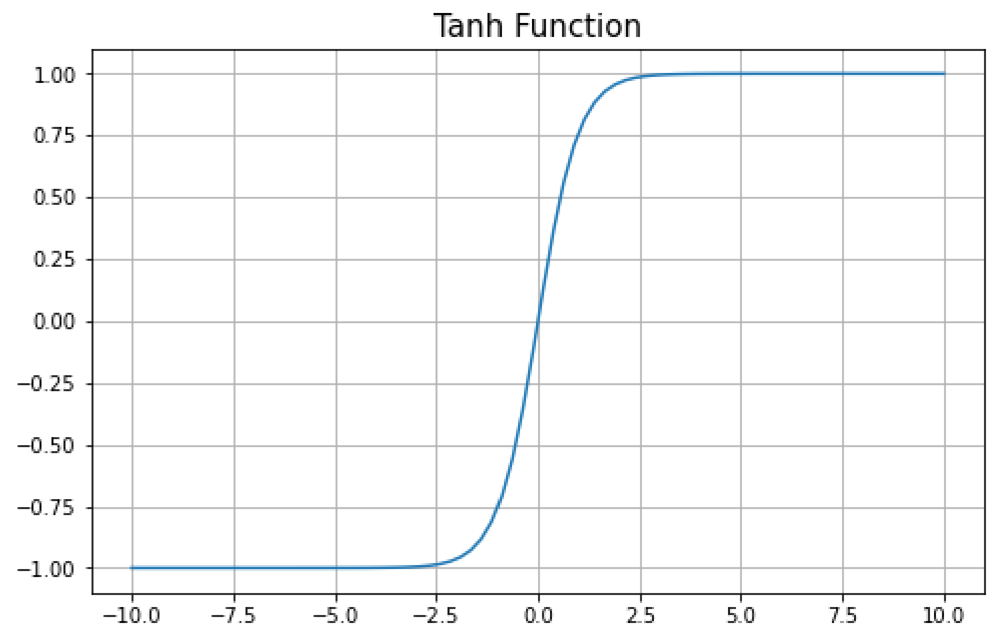
It is very similar to the sigmoid except that the output values are in the range of -1 to +1.

Tanh is said to be **zero centered.**

It is **Differentiable** and **Monotonic** (its derivative is not monotonic)

It is mainly used classification between two classes.

The difference between the sigmoid and tanh is that the gradients are not restricted to move in one direction for tanh.



Tanh is likely to converge faster

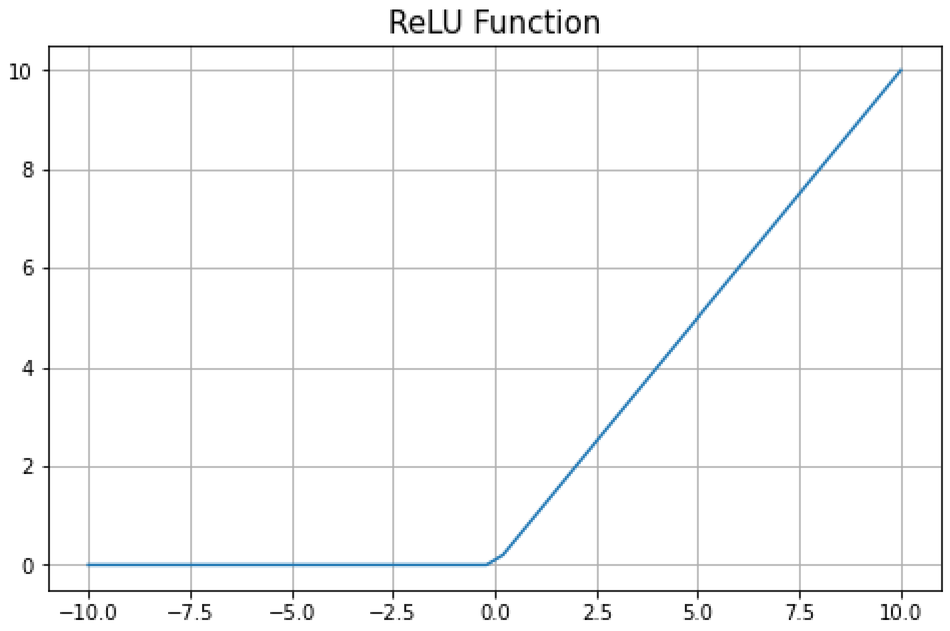
RELU (RECTIFIED LINEAR UNIT)

This activation function is only interested in the POSITIVE VALUES. The output range is from 0 to infinite.

Keeps the input values greater than 0 as is. All the input values less than zero become 0.

The function and its derivative **both are monotonic**.

It is used in almost all the convolutional neural networks or deep learning.



Applying a ReLU function to the output of a neuron, all the values returned become 0

ReLU allows cancelling, but also activate, some of the neurons.

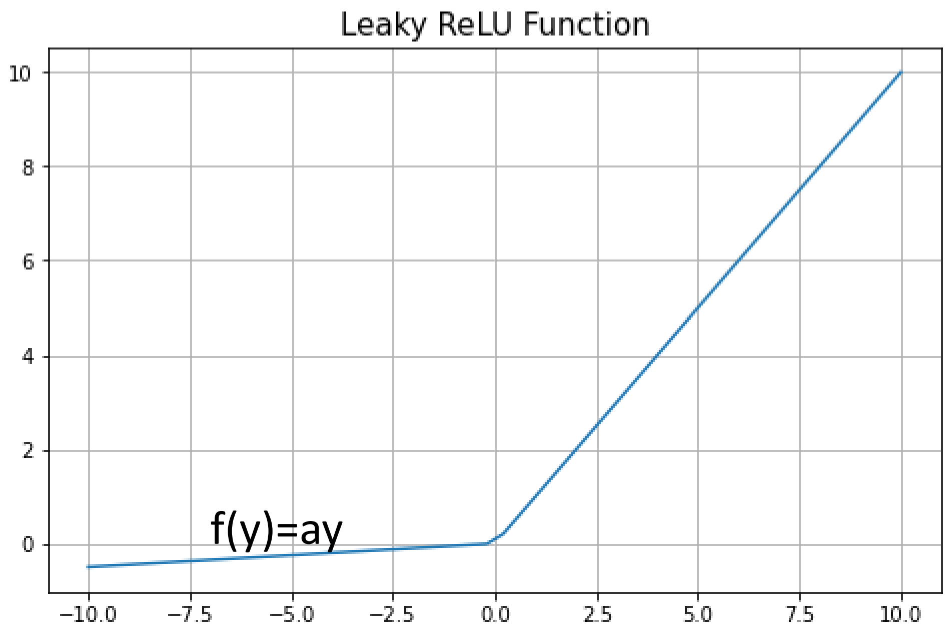
LEAKY RELU

Leaky ReLU outputs a small value for negative inputs. The output values can be from –infinite to infinite.

The leak helps to increase the range of the ReLU function. Usually, the value of a is 0.01 or so.

When **a is not 0.01** then it is called **Randomized ReLU**.

Both Leaky and Randomized ReLU functions are monotonic as well as their derivates.



In general, the desired properties of an activation function are:

Computationally inexpensive

Zero centered

Differentiable.
The derivative of an activation function needs to carry information about the input values because weights are updated based on the gradients.

Not causing vanishing gradient problem

Perceptron was introduced by Frank Rosenblatt in 1957.

It is the basic unit of a neural network (an artificial neuron).

It is a single layer binary linear classifier commonly used to classify the data into two parts.

It is used in supervised learning.

A linear decision boundary is drawn, allowing the distinction between the two linearly separable classes. If the sum of the input signals exceeds a certain threshold, it outputs a signal; otherwise, there is no output.

Perceptron consists of **five** parts:

1. N **inputs**, $x_1 \ldots x_n$
2. **Weights** for each input, $w_1 \ldots w_n$
3. A **bias** input x_0 (constant) and associated weight w_0
4. **Weighted sum of inputs**, $y = w_0x_0 + w_1x_1 + \ldots + w_nx_n$
5. A **threshold** or **activation function**

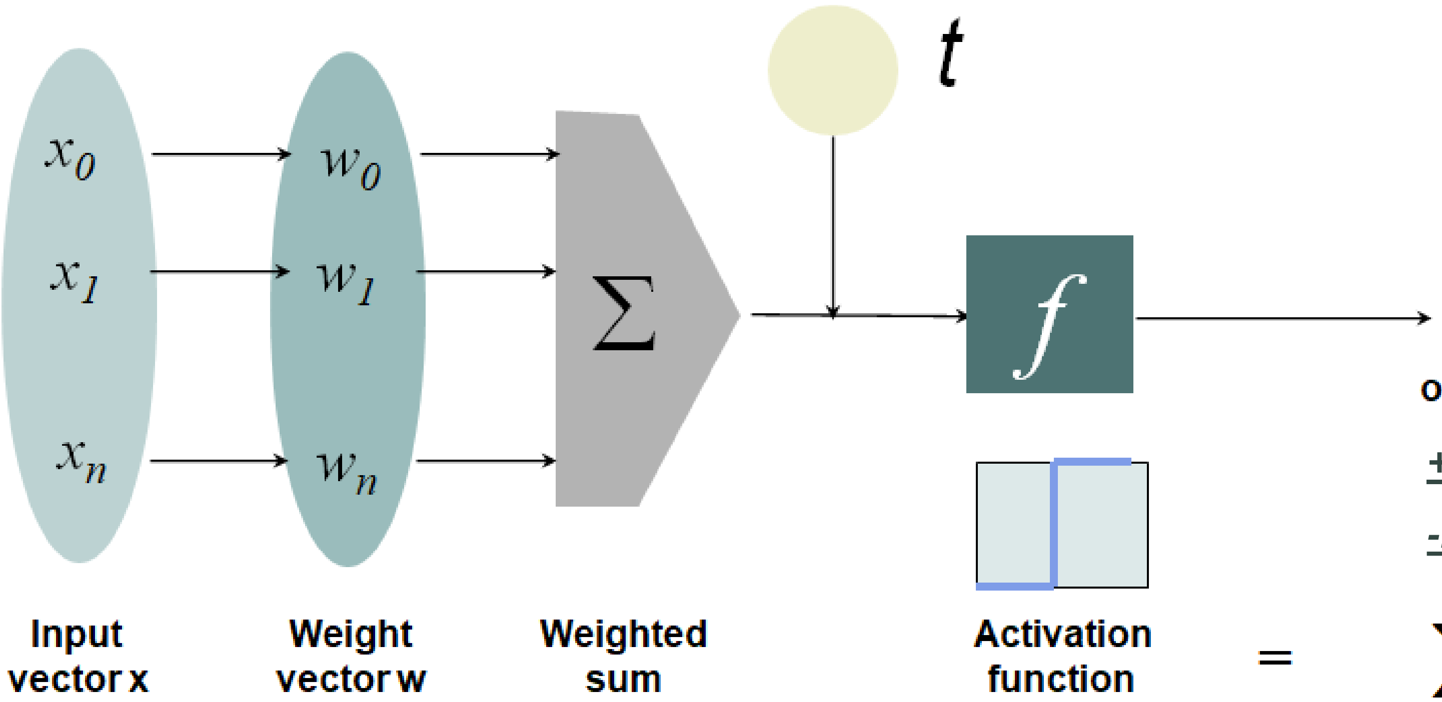
    1 if $y > t$

    0 or -1 if $y <= t$

The Perceptron algorithm **learns** the **weights** for the **input signals** in order to draw a **linear decision boundary** that allows to **distinguish** between the **two linearly separable classes**.

Activation functions are used to map the input between the required values, i.e (0, 1) or (-1, 1), depending on which activation function is used.

The *n*-dimensional <u>input vector</u> **x** is mapped into variable **y** by means of the scalar product and a nonlinear function mapping.



output *y*

<u>+1 output</u> *-> the* ***neuron*** *has been* ***triggered***

<u>-1 or 0 output</u> *-> the neuron was* ***not triggered***

$$= \sum w_i x_i - t$$

1. All the inputs ***x*** are **multiplied** with their weights ***w***.

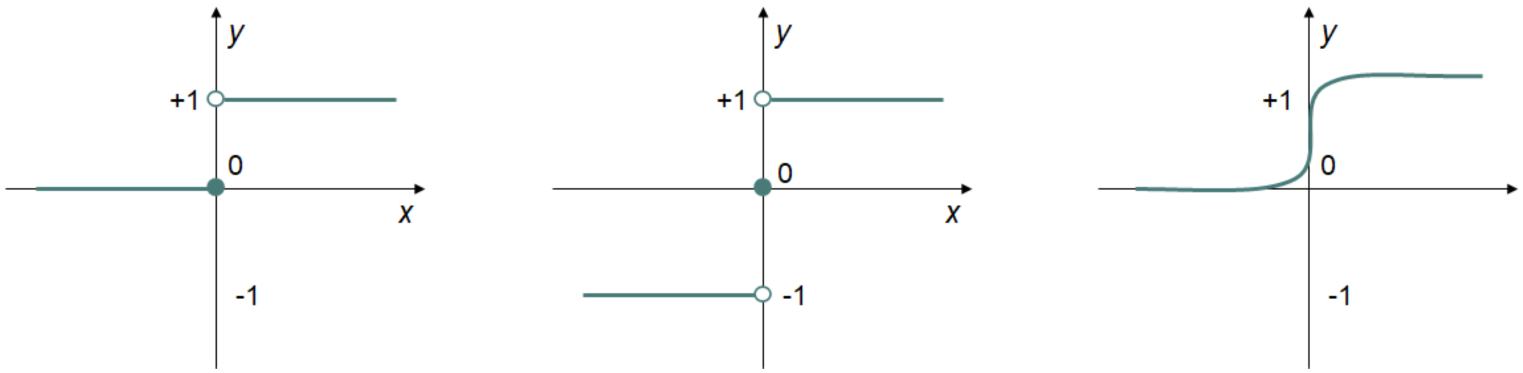
2. **Add** all the multiplied values — **Weighted Sum.**

3. **Add** the **bias** or **threshold** value to the Weighted Sum.

   A bias adjusts the boundary away from the origin without any dependence on the input value, allowing to shift the activation function curve up or down.

4. **Apply** that weighted sum to the correct **Activation Function**.

The **step**, **sign** and **sigmoid** functions are examples of activation functions.



Step Function Sign Function Sigmoid Function

5. In Perceptron, the **predicted output** is **compared** with the **known output**. If it **<u>does not match</u>**, the **error** is propagated **backward** to allow **weight adjustment**.

Optimal weight coefficients are **automatically** learned.

- Perceptron will always converge if the data is separable by a hyperplane.

- Perceptron only works with linearly separable classes and fails to solve non-linear problems. Many challenging AI problems are not linearly separable and thus the Perceptron was discovered to have critical weakness.

- In the case of modeling logic gates, for example, some scientists discover and claim that Perceptron cannot even learn the exclusive-or problem (XOR) since it is not linearly separable.

Perceptron limitations do not apply to feed-forward networks with intermediate or hidden nonlinear units.

ARTIFICIAL NEURAL NETWORK ARCHITECTURE

An architecture is a family $\mathcal{N} = \{\eta\}$ of networks having the same directed graph and node functions but with possibly different weights on the links.

Based on their network topology, ANN’s can be segmented into **five types** of networks.

- Feed-Forward Neural Netwoks
- Recurrent Neural Networks
- Convolutional Neural Networks
- Deconvolutional Neural Networks
- Modular Neural Networks

An ANN is **feed-forward** if there exists an **ordering of neurons** such that every neuron is only connected to a neuron further down the ordering, i.e., there is only one directional signal flow.

It is the simplest type of neural network.

This type of network does not have cycles, data flows unidirectionally from input to output.

This neural network may or may not have hidden layers.

Feed-forward networks have a **front propagated wave** and **no backpropagation** usually by employing a **classifying activation function**.

The architecture of a **feed-forward network** is defined by a **directed and acyclic graph** and the choice of **node functions**.

A directed acyclic graph is one in which no node has a directed path away from it and back to it - there are no closed "cycles."

A directed, acyclic graph can be levelized into layers:

i) The initial or zeroth layer L_o contains the input nodes. These nodes have no incoming links attached to them.

ii) The first layer L_1 consists of those nodes with inputs provided by links from L_o nodes. The ith layer L_i consists of those nodes having inputs provided by links from L_j nodes for $j < i$.

iii) The final layer contains the output nodes.

FEED-FOWARD NEURAL NETWORKS (FFNN)

Feed-forward networks with hidden nonlinear units are **universal approximators**, capable of approximating any bounded continuous function with arbitrarily small errors.

These types of Neural Networks are **responsive** to **noisy data** and **easy to maintain**.

Feedforward neural networks are used in computer vision and speech recognition when classifying target classes is difficult.

FEED-FOWARD NEURAL NETWORKS (FFNN) APPLICATIONS

Supervised Learning

Data is neither sequential nor time-dependent

Responsive to Noisy Data

Computer Vision and Speech Recognition

An important special case of an **FFNN** is the **Multilayer Perceptron (MLP)**.

This type of network is derived from the Perceptron.

- As mentioned, the Perceptron algorithm only solves linearly-separable classes.
- It is based on a *treshold* function, which is not the most suitable for these types of problems.
- A better solution to the problem of learning weights is to use **standard optimization techniques**.

- In this case, an **error function** is used, which is expressed in terms of the neural network output. The goal of the network then becomes to find the values for the **weights** such that the error function is at its minimum value.

- **Gradient descent techniques** can then be used to determine the **impact** of the **weights** on the value of the **error function**. The **error function** must be **differentiable**, which means it should be **continuous.** The threshold function is not continuous, and so is unsuitable.

- When a function is **differentiable**, it is possible to develop a means of adjusting the weights in a perceptron over as many layers as may be necessary.

DRIVES
Development and Research on Innovative Vocational Education Skills

MLP is the most commonly used feedforward architecture.

In this type of NN, the links to the ith layer L_i come **only** from the **immediately preceding** layer L_{i-1}.

1. Initialise the network, with all weights set to random numbers between -1 and +1.

2. Present the first training pattern and obtain the output.

3. Compare the network output with the target output.

4. Propagate the error backwards.

(i) Correct the output layer of weights using the formula:

$$w_{ho} = w_{ho} + (\eta \delta_o o_h)$$

Where:

- w_{ho} is the weight connecting hidden unit *h* with output unit *o.*
- η is the learning rate.
- o_h is the output at hidden unit *h*.
- δ_o is given by: $\delta_o = o_o(1 - o_o)(t_o - o_o)$

o_o is the output at node *o* of the output layer,and *t−o* is the target output for that node

(ii) Correct the input weights using the formula:

$$w_{ih} = w_{ih} + (\eta \delta_h o_i)$$

Where:

- w_{ih} is the weight connecting node i of the input layer with node h of the hidden layer.
- o_i is the input at node i of the input layer, η is the learning rate.
- δ_h is given by: $\delta_h = o_h(1 - o_h) \sum_o (\delta_o w_{ho})$

5. Calculate the **error**, by taking the average difference between the target and the output vector.

6. Repeat the process from step 2 for **each pattern** in the training set to complete **one epoch**.

7. Shuffle the **training set randomly**, to prevent the network from being influenced by the order of the data.

8. Repeat the process from step 2 for a set number of epochs, or **until** the **error ceases to change**.

A **recurrent** NN, also known as **feedback** NN, works on the principle of **saving** the **output** of a layer and **feeding** it **back** to the **input** to help predict the outcome of the layer.

The output neurons can be connected to their inputs.

The feedback network feeds information back into itself to achieve the best-evolved results internally.

Signals in this type of ANN can continuously circulate.

A **Recurrent** network is a neural network with feedback (closed loop) connections.

Examples: BAM, Hopfield, Boltzmann machine, and recurrent backpropagation networks

RECURRENT NEURAL NETWORKS (RNN)

The first layer is formed similar to the feed-forward neural network with the **product** of the **sum of the weights** and the **features**. The recurrent neural network process starts once this is computed, which means that from one time step to the next each neuron will remember some information it had in the previous time-step.

Thus, **each neuron** acts like a **memory cell** in performing computations. In this process, the neural network works on the **front propagation** and remembers what information it needs for later use. If the prediction is **wrong,** the **learning rate** or **error correction** is used to make small changes so that it will **gradually** work towards making the **right** prediction during the **back propagation**.

The architectures range from **fully interconnected** to **partially connected** networks.

Fully connected networks do not have distinct input layers of nodes, and each node has input from all other nodes. Feedback to the node itself is possible.

Simple partially recurrent neural networks have been used to learn strings of characters. Athough some nodes are part of a feedforward structure, other nodes provide the sequential context and receive feedback from other nodes.

DRIVES
Development and Research on Innovative Vocational Education Skills

- Sequence Prediction Problems
- Time Series Prediction
- Natural Language Processing
- Video Tagging

Convolutional Neural Networks (CNNs) are analogous to traditional ANNs in that they are made up of neurons that **self-optimise** through learning. Each neuron receives an input and performs an operation - the basis of countless ANNs.

CNNs are similar to FFNNs, where the neurons have learnable weights and biases.

CNNS are designed to take advantage of images (2D).

Its application has been in signal and image processing which takes over OpenCV in the field of computer vision.

The input layer of a **Convolutional** network will hold the pixel values of the image, and the last layer will contain loss functions associated with the classes.

Convolutional neural networks are similar to feed forward neural networks. One of the **largest limitations** of traditional forms of ANN is that they tend to **struggle** with the **computational complexity** required to **compute image data**.

The only notable difference between **Convolutional Neural Networks** (**CNNs)** and traditional ANNs is that **CNNs** are primarily used in the field of **pattern recognition** within **images**. This allows us to **encode** image-specific features into the architecture, making the network more suited for **image-focused tasks** - whilst further reducing the parameters required to set up the model.

DRIVES
Development and Research on Innovative Vocational Education Skills

CNNs are comprised of **three** types of layers:

CONVOLUTIONAL LAYERS

The output of neurons connected to local regions of the input will be determined by the convolutional layer by calculating the scalar product between their weights and the region connected to the input volume.

POOLING LAYERS

The pooling layer will then simply perform **downsampling** along the **spatial dimensionality** of the given input, **reducing** the **number of parameters** within that activation even further.

FULLY-CONNECTED LAYERS

The fully-connected layers will then perform the same functions as standard ANNs and attempt to produce **class scores** from the **activations** for the classification.

ReLu may be used between these layers to improve performance.

When these layers are stacked, a CNN architecture has been formed.

Through this simple method of transformation, **CNNs** are able to **transform** the **original input layer by layer** using **convolutional** and **downsampling** techniques to produce **class scores** for **classification** and **regression** purposes.

Computer vision techniques are dominated by **convolutional neural networks** because of their **accuracy** in **image classification**.

CONVOLUTIONAL NEURAL NETWORKS (CNN)

Computer Vision

Facial Recognition

OCR

Image Analysis

SUMMARY

- Neurons send and receive signals from your brain.
- There are 3 types of neurons: Sensory Neurons, Motor Neurons and Interneurons. Each one has specific functions like receive, integrate and communicate signals.
- The artificial neuron is a **simplified and simulated model** of the real neuron as well as its basic characteristics.
- In a Neural Network, the neurons are arranged into multiple layers (1 Input Layer, n Hidden Layers, 1 Output Layer).
- There are manu simmilarities between the biological neurons and the artificial ones(both use electrical signals to send messages, have memory that can grow and can adapt and learn). But there are also many differences like the limited size of the artificial neural network, the speed or even the fault tollerance that neural networks can't replicate.
- The output of a neural network is determined by activation function. The most used are: Sigmoid Function, Tanh Function, ReLu and Leaky ReLu.
- **The optimal activation function should be zero centered, use low computational resources, not caus vanishing gradient problem and be differenciable.**
- An Artificial Neural Network is an information processing paradigm that is inspired by the biological nervous systems, such as the human brain’s information processing mechanism.

SUMMARY

- Deep Learning is a subfield of machine learning based on algorithms inspired in artificial neural networks with many layers of nodes, that are capable of identify what something is.
- Artificial intelligence, machine learning and deep learning have many differences but also have similarities.
- An important special case of an FFNN is the Multilayer Perceptron (MLP), in which an error function is used, which is expressed in terms of the neural network output. The goal of the network then becomes to find the values for the weights such that the error function is at its minimum value.
- An RNN, also known as Feedback NN, feeds the output of a layer back to the input to help predict the outcome of the layer, i.e., signals in this type of ANN can continuously circulate.
- Convolutional Neural Networks (CNNs) are similar to feed forward neural networks but are more suited for image-focused tasks. They are comprised of three types of layers: convolutional layers, pooling layers and fully-connected layers. ReLu may be used between these layers to improve performance.

REFFERENCES

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Albawi, S., Mohammed, T. A., & Al-Zawi, S.(2017, August). Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)* (pp. 1-6). Ieee.

Changeux, J. P. (1997). *Neuronal man: The biology of mind* (Vol. 21). Princeton University Press.

Deep learning for smart agriculture: Concepts, tools, applications, and opportunities. *International Journal of Agricultural and Biological Engineering*, *11*(4), 32-44.

Eluyode, O. S., & Akomolafe, D. T. (2013). Comparative study of biological and artificial neural networks. *European Journal of Applied Engineering and Scientific Research*, *2*(1), 36-46.

Fine, T. L. (2006). *Feedforward neural network methodology*. Springer Science & Business Media.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1, No. 2). Cambridge: MIT press.

Gupta, N. (2013). Artificial neural network. *Network and Complex Systems*, *3*(1), 24-28.

Harmon, L. D. (1959). Artificial neuron. *Science*, *129*(3354), 962-963.

Ito, Y. (1991). Representation of functions by superpositions of a step or sigmoid function and their applications to neural network theory. *Neural Networks*, *4*(3), 385-394.

REFFERENCES

Kalman, B. L., & Kwasny, S. C. (1992, June). Why tanh: choosing a sigmoidal function. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks* (Vol. 4, pp. 578-581). IEEE.

Krenker, A., Bešter, J., & Kos, A. (2011). Introduction to the artificial neural networks. *Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech*, 1-18.

Levitan, I. B., Levitan, I. B., & Kaczmarek, L. K. (2002). *The neuron: cell and molecular biology*. Oxford University Press, USA.

Lin, J. W. (2017). Artificial neural network related to biological neuron network: a review. *Advanced Studies in Medical Sciences*, *5*(1), 55-62.

Medsker, L. R., & Jain, L. C. (2001). Recurrent neural networks. *Design and Applications*, *5*.

Nicholls, J. G., Martin, A. R., Wallace, B. G., & Fuchs, P. A. (2001). *From neuron to brain* (Vol. 271). Sunderland, MA: Sinauer Associates.

Noriega, L. (2005). Multilayer perceptron tutorial. *School of Computing. Staffordshire University*.

https://analyticsindiamag.com/6-types-of-artificial-neural-networks-currently-being-used-in-todays-technology/

O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

REFFERENCES

Sharma, S. (2017). Activation functions in neural networks. *towards data science*, *6*.

Sibi, P., Jones, S. A., & Siddarth, P. (2013). Analysis of different activation functions using back propagation neural networks. *Journal of theoretical and applied information technology*, *47*(3), 1264-1268.

Specht, D. F. (1991). A general regression neural network. *IEEE transactions on neural networks*, *2*(6), 568-576.

Svoboda, K. (2011). The past, present, and future of single neuron reconstruction. *Neuroinformatics*, *9*(2-3), 97.

Wang, S. C. (2003). Artificial neural network. In *Interdisciplinary computing in java programming* (pp. 81-100). Springer, Boston, MA.

Zhang, X., Zou, Y., & Shi, W. (2017, August). Dilated convolution neural network with LeakyReLU for environmental sound classification. In *2017 22nd International Conference on Digital Signal Processing (DSP)* (pp. 1-5). IEEE.

https://www.nature.com/articles/d41586-018-01290-0

Zhu, N., Liu, X., Liu, Z., Hu, K., Wang, Y., Tan, J., ... & Guo, Y. (2018).

REFERENCE TO AUTHORS

Regina Sousa

- PhD student in Biomedical Engineering
- Research Collaborator of the Algoritmi Research Center

iD 0000-0002-2988-196X

Diana Ferreira

- PhD student in Biomedical Engineering
- Research Collaborator of the Algoritmi Research Center

iD 0000-0003-2326-2153

Ana Luísa Sousa

- PhD student in Information System and Tecnologies
- Research Collaborator of the Algoritmi Research Center

iD 0000-0001-5731-3583

António Abelha

- Assistant Professor at the University of Minho
- Integrated Researcher of the Algoritmi Research Center

0000-0001-6457-0756

José Machado

- Associate Professor with Habilitation at the University of Minho
- Integrated Researcher of the Algoritmi Research Center

0000-0003-4121-6169

Victor Alves

- Assistant Professor at the University of Minho
- Integrated Researcher of the Algoritmi Research Center

0000-0003-1819-7051

REFERENCE TO AUTHORS

This Training Material has been certified according to the rules of **ECQA – European Certification and Qualification Association.**

The Training Material was developed within the international job role committee **"Computer Vision Expert":**

UMINHO – University of Minho (https://www.uminho.pt/PT)

The development of the training material was partly funded by the EU under Blueprint Project DRIVES.

DRIVES

Development and Research on Innovative Vocational Education Skills

Thank you for your attention

DRIVES project is project under **The Blueprint for Sectoral Cooperation on Skills in Automotive Sector**, as part of New Skills Agenda.

The aim of the Blueprint is **to support an overall sectoral strategy and to develop concrete actions to address short and medium term skills needs.**

Follow DRIVES project at:

More information at:

www.project-drives.eu

Co-funded by the Erasmus+ Programme of the European Union

The Development and Research on Innovative Vocational Educational Skills project (DRIVES) is co-funded by the Erasmus+ Programme of the European Union under the agreement 591988-EPP-1-2017-1-CZ-EPPKA2-SSA-B. The European Commission support for the production of this publication does not constitute endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.