**What is the Sigmoid Function?**

<https://deepai.org/machine-learning-glossary-and-terms/sigmoid-function>

A Sigmoid function is a mathematical function which has a characteristic S-shaped curve. There are a number of common sigmoid functions, such as the **logistic function**, the **hyperbolic tangent**, and the **arctangent**

*sigmoid function* is normally used to refer specifically to the logistic function, also called the logistic sigmoid function.

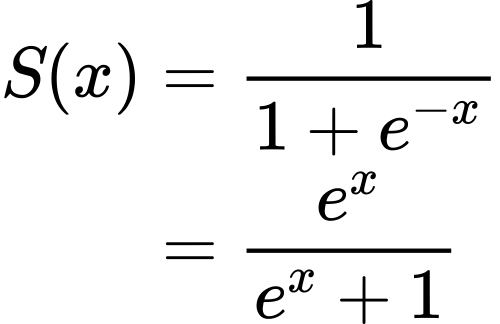
All sigmoid functions have the property that they map the entire number line into a small range such as between 0 and 1, or -1 and 1, so one use of a sigmoid function is to convert a real value into one that can be interpreted as a [probability](https://deepai.org/machine-learning-glossary-and-terms/probability).

Sigmoid functions have become popular in [deep learning](https://deepai.org/machine-learning-glossary-and-terms/deep-learning) because they can be used as an [activation function](https://deepai.org/machine-learning-glossary-and-terms/activation-function) in an artificial [neural network](https://deepai.org/machine-learning-glossary-and-terms/neural-network). They were inspired by the activation potential in biological neural networks.

Sigmoid functions are also useful for many machine learning applications where a real number needs to be converted to a probability. A sigmoid function placed as the last layer of a machine learning model can serve to convert the model's output into a probability score, which can be easier to work with and interpret.

Sigmoid functions are an important part of a [logistic regression](https://deepai.org/machine-learning-glossary-and-terms/logistic-regression) model. Logistic regression is a modification of [linear regression](https://deepai.org/machine-learning-glossary-and-terms/linear-regression) for two-class classification, and converts one or more real-valued inputs into a probability, such as the probability that a customer will purchase a product.

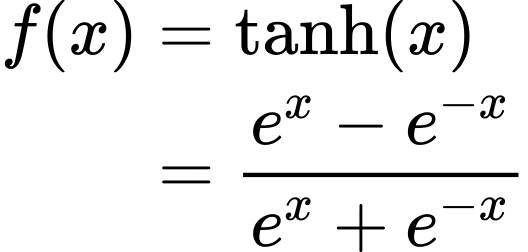
One of the commonest sigmoid functions is the logistic sigmoid function. This is often referred to as *the Sigmoid Function* in the field of machine learning. The logistic sigmoid function is defined as follows:



The logistic function takes any real-valued input, and outputs a value between zero and one.

### **Hyperbolic Tangent Function Formula**

Another common sigmoid function is the hyperbolic function. This maps any real-valued input to the range between -1 and 1.



### **Arctangent Function Formula**

## Sigmoid Function vs. ReLU

In modern artificial neural networks, it is common to see in place of the sigmoid function, the rectifier, also known as the [rectified linear unit](https://deepai.org/machine-learning-glossary-and-terms/rectified-linear-units), or [ReLU](https://deepai.org/machine-learning-glossary-and-terms/relu), being used as the activation function.

The ReLU function has several main advantages over a sigmoid function in a neural network. The main advantage is that the ReLU function is very fast to calculate. In addition, an activation potential in a biological neural network does not continue to change for negative inputs, so the ReLU seems closer to the biological reality if a goal is to mimic biological systems.

In addition, for positive *x*

 the ReLU function has a constant gradient of 1, whereas a sigmoid function has a gradient that rapidly converges towards 0. This property makes neural networks with sigmoid activation functions slow to train. This phenomenon is known as the [vanishing gradient problem](https://deepai.org/machine-learning-glossary-and-terms/vanishing-gradient-problem). The choice of ReLU as an activation function alleviates this problem because the gradient of the ReLU is always 1 for positive

*x* and so the learning process will not be slowed down by the gradient becoming small.

However, the zero gradient for negative *x* can pose a similar problem, known as the zero gradient problem, but it is possible to compensate for this by adding a small linear term in *x* to give the ReLU function a nonzero slope at all points.

## Applications of Sigmoid Function

A key area of machine learning where the sigmoid function is essential is a logistic regression model. A logistic regression model is used to [estimate](https://deepai.org/machine-learning-glossary-and-terms/estimator) the probability of a binary event, such as dead vs alive, sick vs well, fraudulent vs honest transaction, etc. It outputs a probability value between 0 and 1.

In logistic regression, a logistic sigmoid function is fit to a set of data where the independent variable(s) can take any real value, and the dependent variable is either 0 or 1.

### **Why is the logistic function used in logistic regression, and not another sigmoid function?**

The reason that the logistic function is used in logistic regression, and none of the other sigmoid variants, is not just due to the fact that it conveniently returns values between 0 and 1. Logistic regression is derived from the assumption that data in both classes is [normally distributed](https://deepai.org/machine-learning-glossary-and-terms/normal-distribution).

### **Sigmoid function as activation function in artificial neural networks**

An artificial neural network consists of several layers of functions, layered on top of each other:

Each layer typically contains some weights and biases and functions like a small linear regression. A crucial part of the layers is also the activation function.

Razlog za aktivacijske funkcije

However, if every layer in the neural network were to contain only weights and biases, but no activation function, the entire network would be equivalent to a single linear combination of weights and biases. In other words, the formula for the neural network could be factorized and simplified down to a simple linear regression model. Such a model would be able to pick up very simple linear dependencies but unable to perform the impressive tasks that neural networks are renowned for, such as image and voice recognition.

Activation functions were introduced between layers in neural networks in order to introduce a non-linearity. Originally sigmoid functions such as the logistic function, arctangent, and hyperbolic tangent were used, and today ReLU and its variants are very popular. All activation functions serve the same purpose: to introduce a non-linearity into the network. Sigmoid functions were chosen as some of the first activation functions thanks to their perceived similarity with the activation potential in biological neural networks.

Thanks to the use of a sigmoid function at various points within a multi-layer neural network, neural networks can be built to have successive layers pick up on ever more sophisticated features of an input example.

## Sigmoid Function History

In 1798, the English cleric and economist Thomas Robert Malthus published a book under a pseudonym called *An Essay on the Principle of Population*, asserting that the population was increasing in a geometric progression (doubling every 25 years) while food supplies were increasing arithmetically, and that the difference between the two was due to cause widespread famine.

In the late 1830s, the Belgian mathematician Pierre François Verhulst was experimenting with different ways of modeling population growth, and wanted to account for the fact that a population's growth is ultimately self-limiting, and does not increase exponentially forever. Verhulst chose the logistic function as a logical adjustment to the simple exponential model, in order to model the slowing down of a population's growth which occurs when a population begins to exhaust its resources.

In 1972, the biologists Hugh Wilson and Jack Cowan at the University of Chicago were attempting to model biological neurons computationally and published the Wilson–Cowan model, where a neuron sends a signal to another neuron if it receives a signal greater than an activation potential. Wilson and Cowan chose the logistic sigmoid function to model the activation of a neuron as a function of a stimulus.