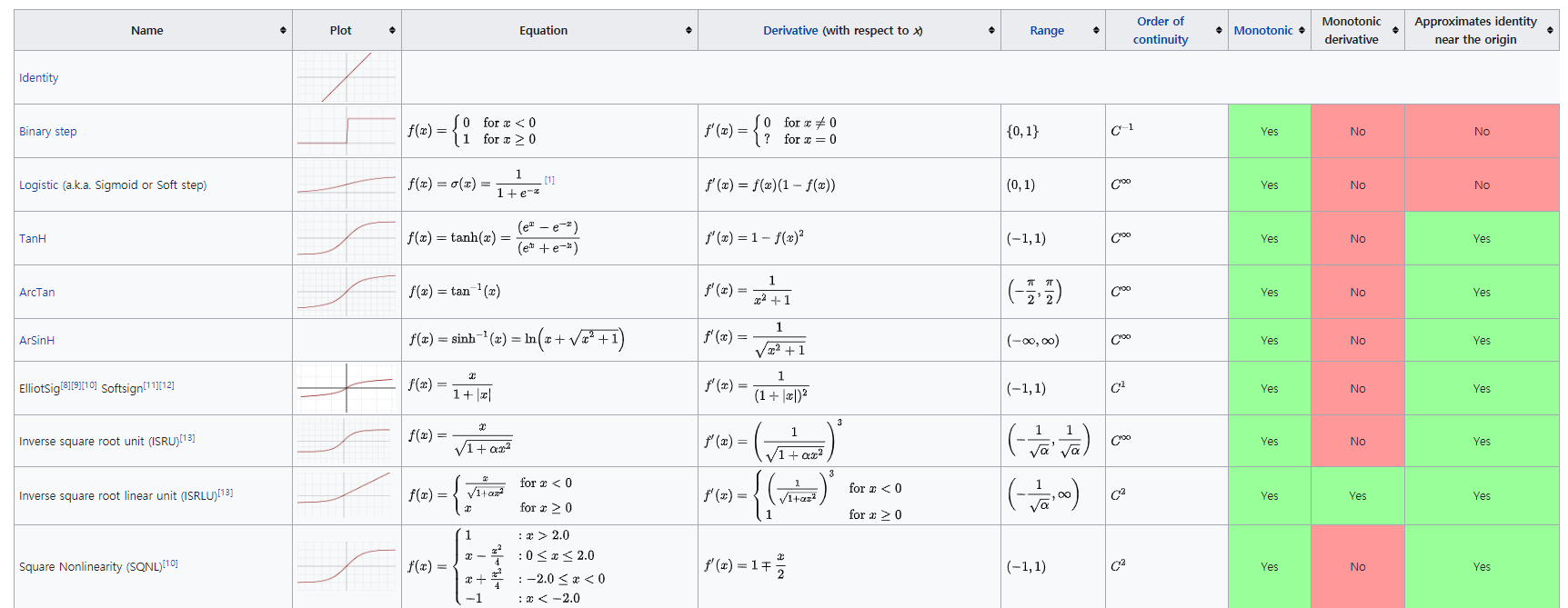
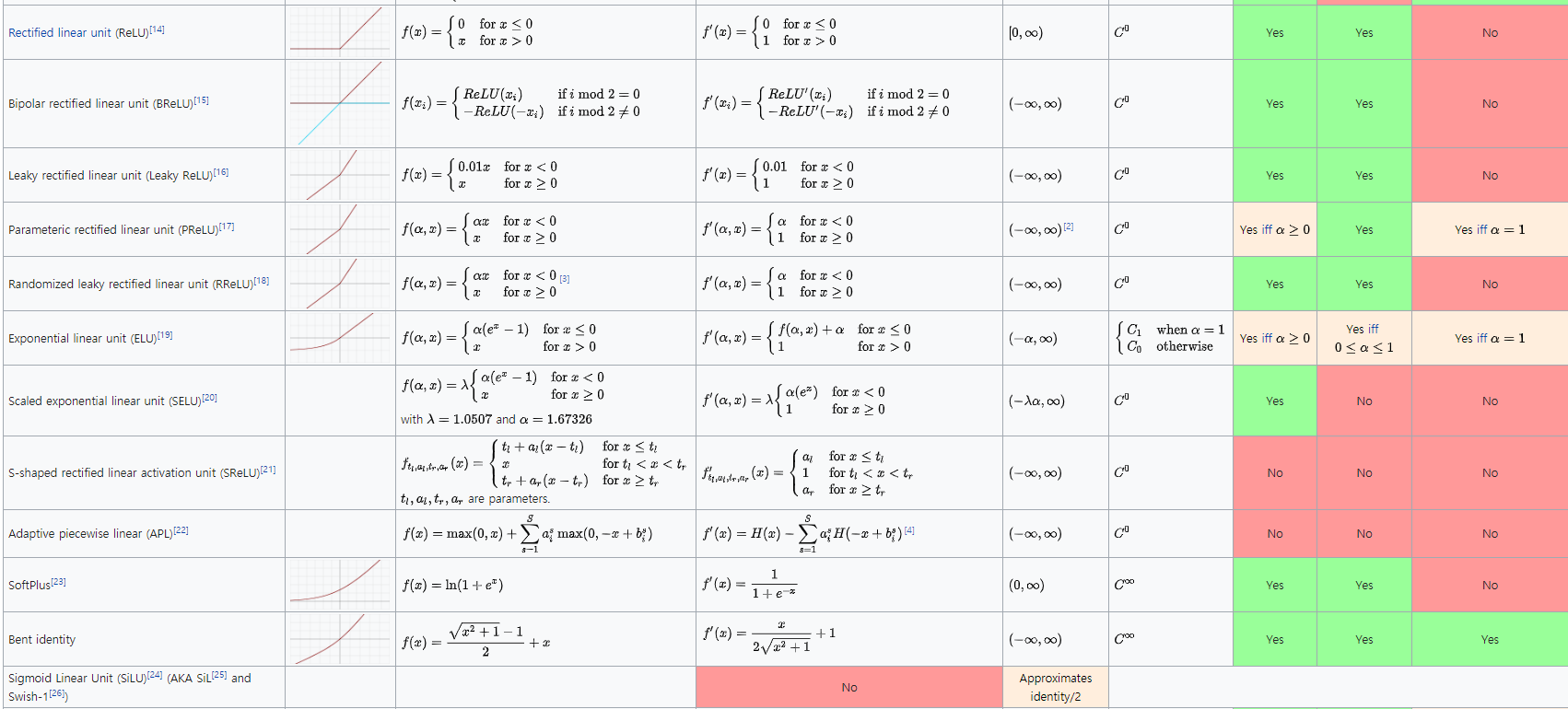
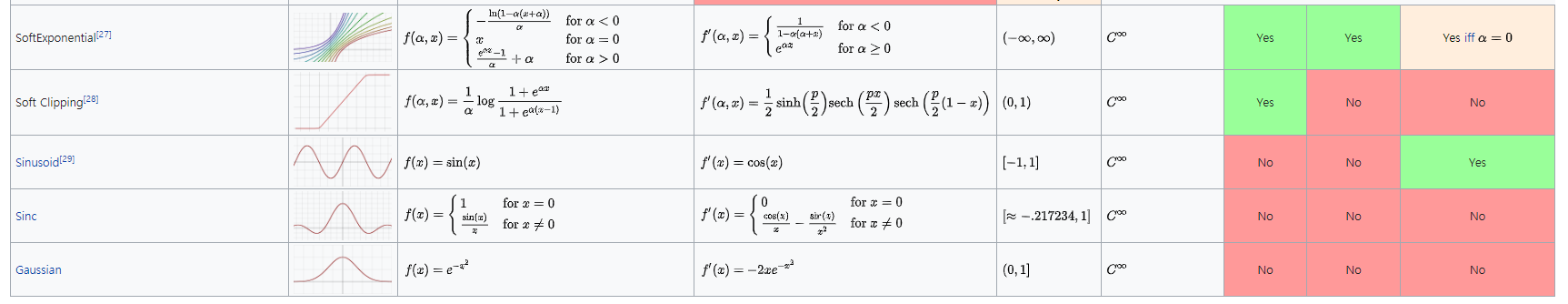
**자주 사용되는 활성화 함수의 종류와 기능**

**목적:** In [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network), the activation function of a node defines the output of that node given an input or set of inputs. A standard [computer chip circuit](https://en.wikipedia.org/wiki/Integrated_circuit) can be seen as a [digital network](https://en.wikipedia.org/wiki/Digital_electronics) of activation functions that can be "ON" (1) or "OFF" (0), depending on input. This is similar to the behavior of the [linear perceptron](https://en.wikipedia.org/wiki/Linear_perceptron) in [neural networks](https://en.wikipedia.org/wiki/Neural_networks). However, only *nonlinear* activation functions allow such networks to compute nontrivial problems using only a small number of nodes.[[1]](https://en.wikipedia.org/wiki/Activation_function#cite_note-1) In [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network), this function is also called the [transfer function](https://en.wikipedia.org/wiki/Transfer_function)

- 간단히 말해서 생물학에서 세포내의 action potential (firing some action)의 기능을 하는 함수

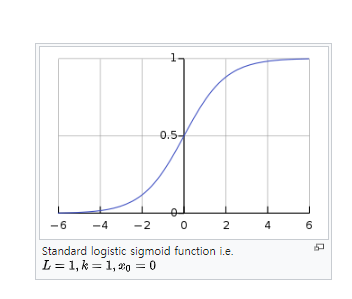


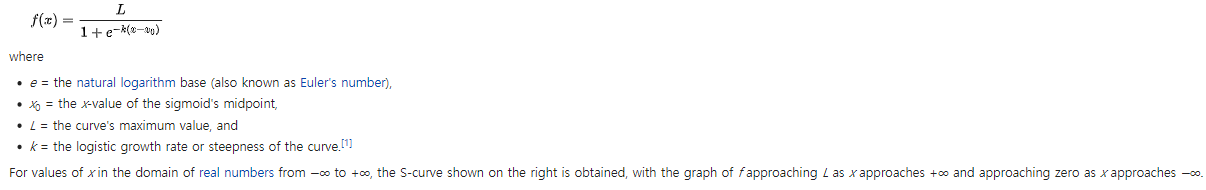




<https://en.wikipedia.org/wiki/Activation_function>

**1) Logistic function**





**In statistics and machine learning**

Logistic functions are used in several roles in statistics. For example, they are the cumulative distribution function(누적 확률 분포) of the logistic family of distribution, and they are, a bit simplified, used to model the chance a chess player has to beat his opponent in the Elo rating system. More specific examples now follow

[로지스틱 회귀의 목적은 일반적인 [회귀 분석](https://ko.wikipedia.org/wiki/%ED%9A%8C%EA%B7%80_%EB%B6%84%EC%84%9D)의 목표와 동일하게 [종속 변수](https://ko.wikipedia.org/wiki/%EB%8F%85%EB%A6%BD_%EB%B3%80%EC%88%98%EC%99%80_%EC%A2%85%EC%86%8D_%EB%B3%80%EC%88%98)와 독립 변수간의 관계를 구체적인 함수로 나타내어 향후 예측 모델에 사용하는 것이다. 이는 독립 변수의 선형 결합으로 종속 변수를 설명한다는 관점에서는 [선형 회귀](https://ko.wikipedia.org/wiki/%EC%84%A0%ED%98%95_%ED%9A%8C%EA%B7%80) 분석과 유사하다. 하지만 로지스틱 회귀는 [선형 회귀](https://ko.wikipedia.org/wiki/%EC%84%A0%ED%98%95_%ED%9A%8C%EA%B7%80) 분석과는 다르게 종속 변수가 범주형 데이터를 대상으로 하며 입력 데이터가 주어졌을 때 해당 데이터의 결과가 특정 분류로 나뉘기 때문에 일종의 분류 ([classification](https://en.wikipedia.org/wiki/classification)) 기법으로도 볼 수 있다. 흔히 로지스틱 회귀는 종속변수가 이항형 문제(즉, 유효한 범주의 개수가 두개인 경우)를 지칭할 때 사용된다. 이외에, 두 개 이상의 범주를 가지는 문제가 대상인 경우엔 다항 로지스틱 회귀 ([multinomial logistic regression](https://en.wikipedia.org/wiki/multinomial_logistic_regression)) 또는 분화 로지스틱 회귀 (**polytomous logistic regression**)라고 하고 복수의 범주이면서 순서가 존재하면 서수 로지스틱 회귀 ([ordinal logistic regression](https://en.wikipedia.org/wiki/ordinal_logistic_regression)) 라고 한다.[[2]](https://ko.wikipedia.org/wiki/%EB%A1%9C%EC%A7%80%EC%8A%A4%ED%8B%B1_%ED%9A%8C%EA%B7%80#cite_note-wal67est-2) 로지스틱 회귀 분석은 의료, 통신, [데이터마이닝](https://ko.wikipedia.org/wiki/%EB%8D%B0%EC%9D%B4%ED%84%B0%EB%A7%88%EC%9D%B4%EB%8B%9D)과 같은 다양한 분야에서 분류 및 예측을 위한 모델로서 폭넓게 사용되고 있다.]

1) Logistic regression

Logistic functions are used in logistic regression to model how the probability P of an event may be affected by one or more explanatory variables

Ex) 

Where x is the explanatory variable and a and b are model parameters to be fitted and f is the standard logistic function.

-Logistic regression and other log-linear models are also commonly used in ML. A generalization of the logistic function to multiply input is the softmax activation function, used in multinomial logistic regression.

-Another application of the logistic function is the Rasch model, used in item response theory. In particular, the Rasch model forms a basis for maximum likelihood estimation of the locations of objects or persons on a continuum, based on collections of categorical data.

-In Neural networks

Logistic functions are used in NN to introduce nonlinearity in the model or to clamp signals to within a specified range. A popular [neural net element](https://en.wikipedia.org/wiki/Artificial_neuron) computes a [linear combination](https://en.wikipedia.org/wiki/Linear_combination) of its input signals, and applies a bounded logistic function to the result; this model can be seen as a "smoothed" variant of the classical [threshold neuron](https://en.wikipedia.org/wiki/Perceptron).

A common choice for the activation or "squashing" functions, used to clip for large magnitudes to keep the response of the neural network boundedis

{\displaystyle g(h)={\frac {1}{1+e^{-2\beta h}}}}

which is a logistic function. These relationships result in simplified implementations of [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network) with [artificial neurons](https://en.wikipedia.org/wiki/Artificial_neuron). Practitioners caution that sigmoidal functions which are [antisymmetric](https://en.wikipedia.org/wiki/Odd_functions) about the origin (e.g. the [hyperbolic tangent](https://en.wikipedia.org/wiki/Hyperbolic_tangent)) lead to faster convergence when training networks with [backpropagation](https://en.wikipedia.org/wiki/Backpropagation).

The logistic function is itself the derivative of another proposed activation function, the [softplus](https://en.wikipedia.org/wiki/Softplus).

**-1) Sigmoid function**

A sigmoid function is a [mathematical function](https://en.wikipedia.org/wiki/Function_(mathematics)) having a characteristic "S"-shaped curve or sigmoid curve. Often, *sigmoid function* refers to the special case of the [logistic function](https://en.wikipedia.org/wiki/Logistic_function) shown in the first figure and defined by the formula{\displaystyle S(x)={\frac {1}{1+e^{-x}}}={\frac {e^{x}}{e^{x}+1}}.}

Special cases of the sigmoid function include the [Gompertz curve](https://en.wikipedia.org/wiki/Gompertz_curve) (used in modeling systems that saturate at large values of x) and the [ogee curve](https://en.wikipedia.org/wiki/Ogee_curve) (used in the [spillway](https://en.wikipedia.org/wiki/Spillway) of some [dams](https://en.wikipedia.org/wiki/Dam)). Sigmoid functions have domain of all [real numbers](https://en.wikipedia.org/wiki/Real_number), with return value [monotonically increasing](https://en.wikipedia.org/wiki/Monotonic_function) most often from 0 to 1 or alternatively from −1 to 1, depending on convention.

A wide variety of sigmoid functions including the logistic and [hyperbolic tangent](https://en.wikipedia.org/wiki/Hyperbolic_tangent) functions have been used as the [activation function](https://en.wikipedia.org/wiki/Activation_function) of [artificial neurons](https://en.wikipedia.org/wiki/Artificial_neuron). Sigmoid curves are also common in statistics as [cumulative distribution functions](https://en.wikipedia.org/wiki/Cumulative_distribution_function)(which go from 0 to 1), such as the integrals of the [logistic distribution](https://en.wikipedia.org/wiki/Logistic_distribution), the [normal distribution](https://en.wikipedia.org/wiki/Normal_distribution), and [Student's *t* probability density functions](https://en.wikipedia.org/wiki/Student%27s_t-distribution).

Application : Many natural processes, such as those of complex system [learning curves](https://en.wikipedia.org/wiki/Learning_curve), exhibit a progression from small beginnings that accelerates and approaches a climax over time. When a specific mathematical model is lacking, a sigmoid function is often used.[[3]](https://en.wikipedia.org/wiki/Sigmoid_function#cite_note-3)

The [van Genuchten–Gupta model](https://en.wikipedia.org/wiki/Van_Genuchten%E2%80%93Gupta_model) is based on an inverted S-curve and applied to the response of crop yield to [soil salinity](https://en.wikipedia.org/wiki/Soil_salinity).

Examples of the application of the logistic S-curve to the response of crop yield (wheat) to both the soil salinity and depth to [water table](https://en.wikipedia.org/wiki/Water_table) in the soil are shown in [logistic function#In agriculture: modeling crop response](https://en.wikipedia.org/wiki/Logistic_function#In_agriculture:_modeling_crop_response).

In [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network), sometimes non-smooth functions are used instead for efficiency; these are known as [hard sigmoids](https://en.wikipedia.org/wiki/Hard_sigmoid).

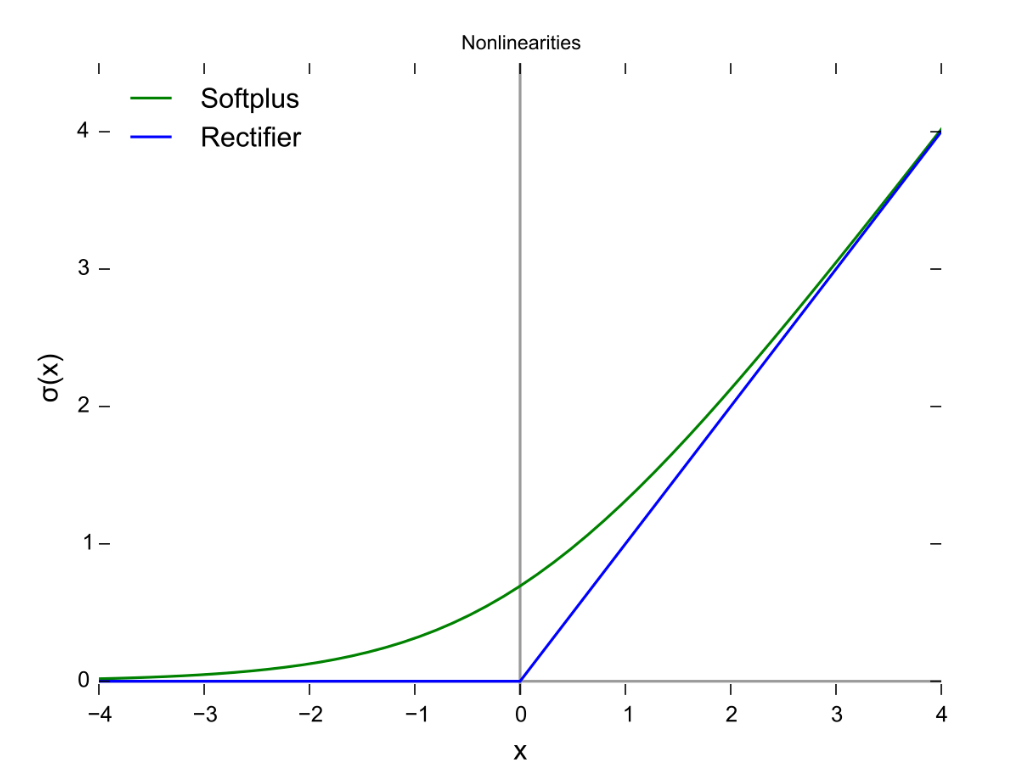
In [biochemistry](https://en.wikipedia.org/wiki/Biochemistry) and [pharmacology](https://en.wikipedia.org/wiki/Pharmacology), the [Hill equation](https://en.wikipedia.org/wiki/Hill_equation_(biochemistry)) and Hill-Langmuir equation are sigmoid functions.

**Examples**

* Logistic function
* Hyperbolic tangent
* Arctangent function
* Gudermannian function
* Error function
* Generalised logistic function
* Smoothstep function
* Some algebraic functions

The integral of any continuous, non-negative, “bump-shaped” function will be sigmoidal, thus the cumulative distribution functions for many common probability distributions are sigmoidal. One such example is the error function, which is related to the cumulative distribution function of a normal distribution.

**2) ReLU (Rectified linear unit)**



In the context of artificial neural network, the rectifier is an activation function defined as the positive part of its argument



Where x is the input to a neuron. This is also known as a ramp function and is analogous to half-wave rectification in electrical engineering. And this function is, as of 2017, the most popular activation for deep neural networks. A unit employing the rectifier is also called a rectified liner unit(ReLU).

Rectified liner units find applications in computer vision and speech recognition using deep neural nets.

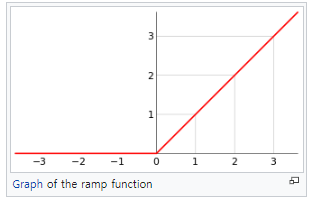
Advantages :

* Biological plausibility : One-sided, compared to the anti-symmetry of tanh
* Sparse activation : For example, in a randomly initialized network, only about 50% of hidden units are activated (having a non-zero output)
* Better gradient propagation : Fewer vanishing gradient problems compared to sigmoidal activation functions that saturate in both directions
* Efficient computation : Only comparison, addition and multiplication.
* Scale-invariant : 

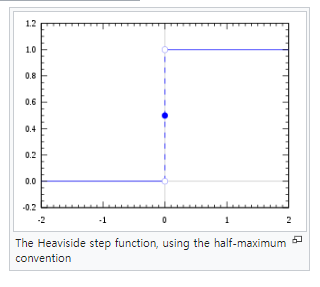
Rectifying activation functions were used to separate specific excitation and unspecific inhibition in the Neural Abstraction Pyramid, which was trained in a supervised way to learn several computer vision tasks. the use of the rectifier as a non-linearity has been shown to enable training deep [supervised](https://en.wikipedia.org/wiki/Supervised_learning) neural networks without requiring [unsupervised](https://en.wikipedia.org/wiki/Unsupervised_learning) pre-training. Rectified linear units, compared to [sigmoid function](https://en.wikipedia.org/wiki/Sigmoid_function) or similar activation functions, allow for faster and effective training of deep neural architectures on large and complex datasets.

Potential Problems :

* Non-differentiable at zero: however, it is differentiable anywhere else, and the value of the derivative at zero can be arbitrarily chosen to be 0 or 1.
* Non-zero centered
* Unbounded
* Dying ReLU problem : ReLU neurons can be pushed into states in which they become inactive for essentially all inputs. In this state, no gradients flow backward through the neuron, and so the neuron becomes stuck in a perpetually inactive state and “dise.” This is a form of the vanishing gradient problem. In some cases, large numbers of neurons in a network can become stuck in data state, effectively decreasing the model capacity. This problem typically arises when the learning rate is set too high. It may be mitigated by using Leaky ReLUs instead, which assign a small positive slope to the left of x = 0.
* Ramp function is a unary real function(A unary function is a [function](https://en.wikipedia.org/wiki/Function_(mathematics)) that takes one [argument](https://en.wikipedia.org/wiki/Parameter_(computer_science))), whose graph is shaped like a ramp. It can be expressed by numerous definitions, for example “0 for negative inputs, output equals input for non-negative inputs”. The term “ramp” can also be used for other functions obtained by scaling and shifting.



* Half-wave rectification : Mathematically, it is a step function(for positive pass, negative block), passing positive corresponds to the ramp function being the identity on positive inputs, blocking negative corresponds to being zero on negative inputs.
* Softplus is a smooth approximation to the rectifier is the analytic function. which is called the softplus or smoothReLU function. The derivative of softplus is the logistic function. The logistic function is a smooth approximation of the derivative of the rectifier, the Heaviside step function.
* Heaviside step function or the unit step function



**1) Noisy ReLUs**

Rectified linear unit can e extended to include Gaussian noise, making them noisy ReLUs, giving



Noisy ReLus have been used with some success in restricted Boltzmann machines for computer vision tasks

**2)** **Leaky ReLUs**

Leaky ReLus allow a small, positive gradient when the unit is not active.



**3) Parametric ReLUs**

Parametric RelUs(PReLUs) take this idea(Leaky ReLUs) further by making the coefficient of leakge into a parameter that is learned along with the other neural network parameters.



Not that for a<= 1, this is equivalent to



And thus has a relation to “maxout” networks.

**4) ELUs**

Exponential linear units try to make the mean activations closer to zero which speeds up learning. It has been shown that ELUs can obtain higher classification accuracy then ReLUs



A is a hyper-parameter to be tuned and a>=0 is a constraint.

**3) Softmax function**

In [mathematics](https://en.wikipedia.org/wiki/Mathematics), the softmax function, also known as softargmaxor normalized exponential function, is a function that takes as input a vector of *K* real numbers, and normalizes it into a [probability distribution](https://en.wikipedia.org/wiki/Probability_distribution) consisting of *K* probabilities. That is, prior to applying softmax, some vector components could be negative, or greater than one; and might not sum to 1; but after applying softmax, each component will be in the [interval](https://en.wikipedia.org/wiki/Interval_(mathematics)) (0,1){\displaystyle (0,1)}, and the components will add up to 1, so that they can be interpreted as probabilities. Furthermore, the larger input components will correspond to larger probabilities. Softmax is often used in [neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network), to map the non-normalized output of a network to a probability distribution over predicted output classes.

The standard (unit) softmax function   {\displaystyle \sigma :\mathbb {R} ^{K}\to \mathbb {R} ^{K}}is defined by the formula



In words : we apply the standard exponential function to each element Zi of the input vector z and normalize these values by dividing by the sum of all these exponentials; this normalization ensures that the sum of the components of the output vector 

Instead of e, a different base b>0 can be used; choosing a larger value of b will create a probability distribution that is move concentrated around the positions of the largest input values.





Applications :

* The softmax function is used in various multiclass classification methods, such as multinomial logistic regression(also known as softmax regression), Multiclass linear discriminant analysis, naïve bates classifiers, and artificial neural networks.
* In neural networks, the softmax function is often used in the final layer of a neural network-based classifier. Such networks are commonly trained under a log loss(or cross-entropy) regime, giving a non-linear variant of multinomial logistic regression. Since the function maps a vector and a specific index I to a real value, the derivative needs to take the index into account

Here, the Kronecker delta is used for simplicity. Multinomial logit for a probability model which uses the softmax activation function

* In Reinforcement learning

A softmax function can be used to convert values into action probabilities. The function commonly used is 

Where the action values  corresponds to the expected reward of following action a and  is called a temperature parameter (in allusion to statistical mechanics). For high temperatures ( 🡪 ∞), all actions have nearly the same probability an the lower the temperature, the more expected rewards affect the probability. For a low temperature  the probability of the action with the highest expected reward tends to 1.