Top ten most useful Activation Function

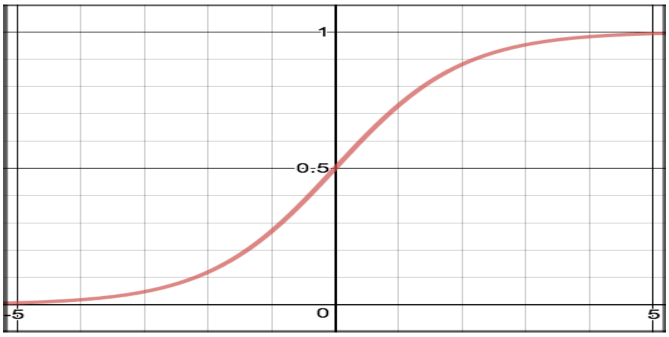
Activation unit calculates the net output of a neural cell neural network. Backpropagation algorithm multiplies the derivative of the activation function. That's why, picked up activation function has to be differentiable. For example, step function is useless in backpropagation because it cannot be backpropagated. That is not a must, but scientists tend to consume activation functions which have meaningful derivatives. Some top ten activation functions are as follows:

1.Sigmoid Function

Sigmoid takes a real value as input and outputs another value between 0 and 1. It's easy to work with and has all the nice properties of activation functions: it's non-linear, continuously differentiable, monotonic, and has a fixed output range.

Function: $S(z)$= 1/1+e−z

It's derivative: $S'(z)$= $S(z)$·(1−$S(z)$)



Pros:

The output of the activation function is always going to be in range (0,1) compared to (-inf, inf) of linear function. So we have our activations bound in a range. Nice, it won't blow up the activations.

Cons:

its output isn't zero centered. It makes the gradient updates go too far in different directions.

It has Vanishing gradient problem, it means after sometime when there is change in the values of x-axis than there is almost no change in y-axis .

2 ReLU Function:

A recent invention which stands for Rectified Linear Units. The formula is deceptively simple: $max(0,z)$max(0,z). Despite its name and appearance, it's not linear and provides the same benefits as Sigmoid but with better performance.

Function : $R(z)$= z if z>0 and 0 if z<0

Its Derivative: R'(z) = 1 if z>0 and 0 if z<0

Pros:
ReLu is less computationally expensive than tanh and sigmoid because it involves simpler mathematical operations.
It avoids and rectifies vanishing gradient problem to some extent

Cons:
One of its limitation is that it should only be used within Hidden layers of a Neural Network Model.
Some gradients can be fragile during training and can die

3 Leaky ReLU Function:
LeakyRelu is a variant of ReLU. Instead of being 0 when z<0z<0, a leaky ReLU allows a small, non-zero, constant gradient αα (Normally, α=0.01α=0.01). However, the consistency of the benefit across tasks is presently unclear.

Function : $R(z)$= z if z>0 and αz if z<=0
Its Derivative: R’(z) = 1 if z>0 and α if z<=0

Pros:
Leaky ReLUs are one attempt to fix the “dying ReLU” problem by having a small negative slope (of 0.01, or so).

Cons:
As it possess linearity, it can’t be used for the complex Classification. It lags behind the Sigmoid and Tanh for some of the use cases.

4 Tanh Function
Tanh squashes a real-valued number to the range [-1, 1]. It’s non-linear. But unlike Sigmoid, its output is zero-centered. Therefore, in practice the tanh non-linearity is always preferred to the sigmoid nonlinearity.

Function: $tanh(z)=e^{z}-e^{-z}$ / $e^{z}+e^{-z}$
Its derivative: $tanh'(z)=1-tanh(z)^{2}$

Pros:
The gradient is stronger for tanh than sigmoid (derivatives are steeper).
Cons:
Tanh also has the vanishing gradient problem.

5 Softmax Activation Function
Softmax function calculates the probabilities distribution of the event over ‘n’ different events. In general way of saying, this function will calculate the probabilities of each target class over all possible target classes

Function:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

Pros:
Good for multiclass classification problems

Cons:
It can be only used for output functions

6 Swish Functions:
Swish is a lesser known activation function which was discovered by researchers at Google. Swish is as computationally efficient as ReLU and shows better performance than ReLU on deeper models. The values for swish ranges from negative infinity to infinity. The function is defined as

f(x) = x*sigmoid(x)
f(x) = x/(1+e^-x)

7 ELU Functions:
Exponential Linear Unit or ELU for short is also a variant of Rectiufied Linear Unit (ReLU) that modifies the slope of the negative part of the function. Unlike the leaky relu and parametric ReLU functions, instead of a straight line, ELU uses a log curve for defning the negatice values. It is defined as

Formula:
f(x) = x,   x>=0
   = a(e^x-1), x<0

8 Maxout Function:
The Maxout activation is a generalization of the ReLU and the leaky ReLU functions. It is a learnable activation function.
It is a piecewise linear function that returns the maximum of the inputs, designed to be used in conjunction with the dropout regularization technique.

One relatively popular choice is the Maxout neuron (introduced recently by Goodfellow et al.) that generalizes the ReLU and its leaky version. Notice that both ReLU and Leaky ReLU are a special case of this form (for example, for ReLU we have w1,b1 =0).
The Maxout activation function is defined as follows:

Function: max($w_1 x^T$ +b1 + $w_2 x^T$ +b2)

Pros:
The Maxout neuron therefore enjoys all the benefits of a ReLU unit (linear regime of operation, no saturation) and does not have its drawbacks
Cons:

However, it doubles the total number of parameters for each neuron and hence, a higher total number of parameters need to be trained.

9 SoftPlus Function:
Softplus is an alternative of traditional functions because it is differentiable and its derivative is easy to demonstrate. Besides, it has a surprising derivative!

The Softplus function is given by: $f(x) = \ln(1+e^{x})$
It"s derivative: *$dy/dx = 1 / (1 + e^{-x})$*

The softplus function is similar to the ReLU function, but it is relatively smooth.It is unilateral suppression like ReLU.It has a wide acceptance range (0, + inf)

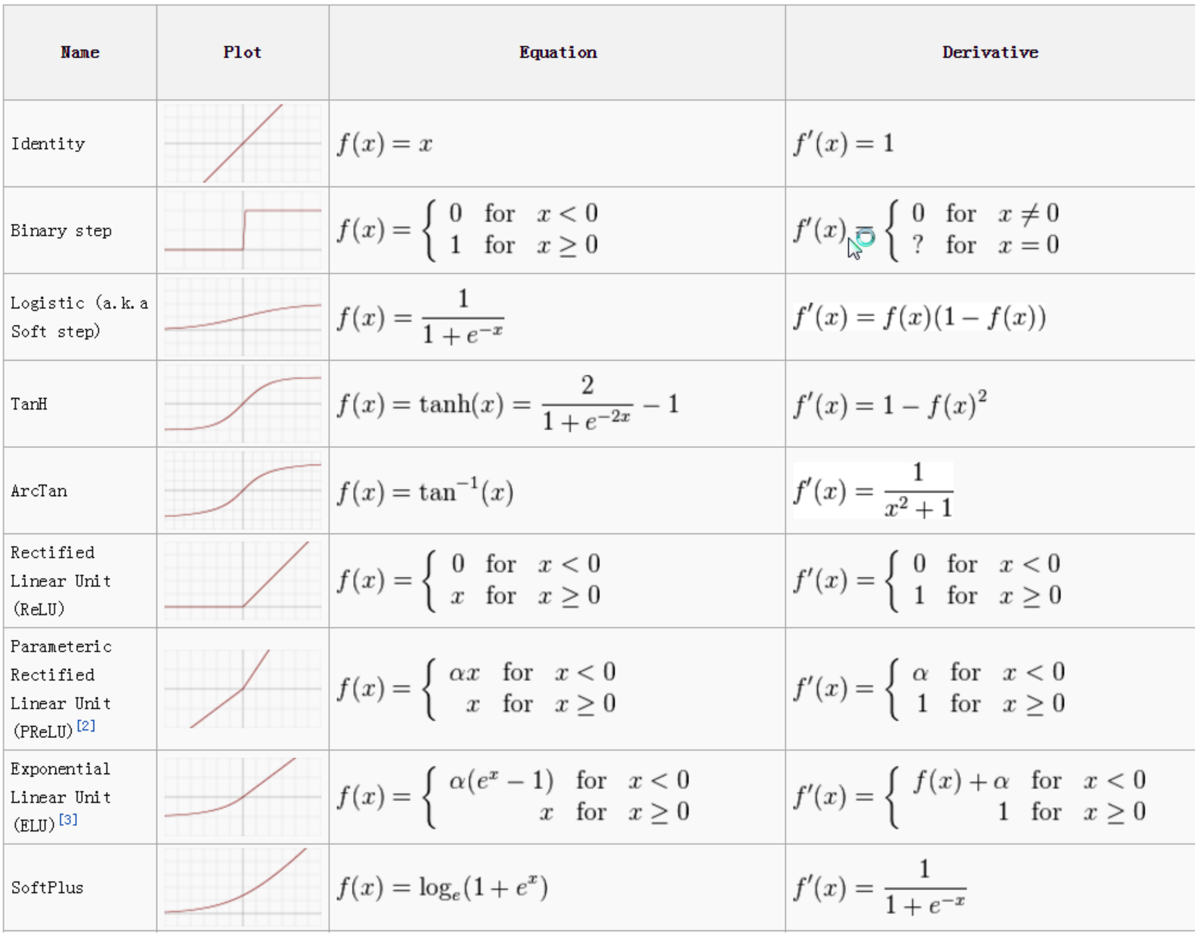
10 Gaussian Activation Functions
Gaussian Activation function comes from the special class of function known as radial basis functions (RBFs) are used in RBF networks. These functions are Bell-Shaped Curves that comes with the properties of having continuous.

Function: $f(x) = e^{-x2}$
Derivative: $f'(x) = -2e^{-x2}$

Pros:
Gaussian functions model probabilities directly (value range between 0 to 1

Cons:
The normal cdf (at least in some implementations/on some platforms) itself will be substantially slower to evaluate that the logistic function. Its derivative won't be much different from the one for the logistic.

In below page there is overview description of most of used activation functions



Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU)[2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU)[3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$