**Activation Functions in Neural Networks**

**=========================ipynb file also present===============**

**What is Activation Function?**

It’s just a thing function that you use to get the output of node. It is also known as **Transfer Function**.

**Why we use Activation functions with Neural Networks?**

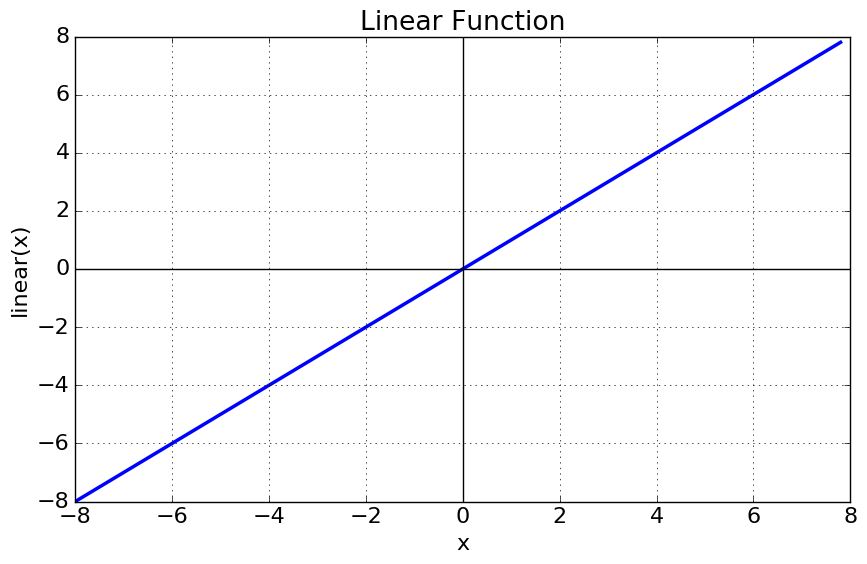
It is used to determine the output of neural network like yes or no, **whether it should take part or not**. It maps the resulting values in between 0 to 1 or -1 to 1 etc. (depending upon the function).

The Activation Functions can be basically divided into 2 types-

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

* **Linear or Identity Activation Function(For regression )**

As you can see the function is a line or linear. Therefore, the output of the functions will not be confined between any ranges.



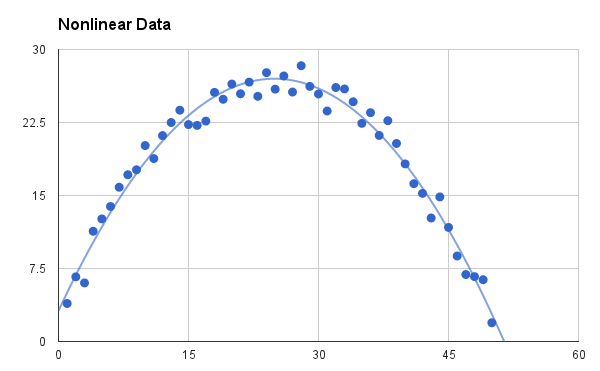
**Equation:**f(x) = x

**Range:** (-infinity to infinity)

It doesn’t help with the complexity or various parameters of usual data that is fed to the neural networks.

* **Non-linear Activation Function**

The Nonlinear Activation Functions are the most used activation functions. Nonlinearity helps to makes the graph look something like this



It makes it easy for the model to generalize or adapt with variety of data and to differentiate between the outputs.

The main terminologies needed to understand for nonlinear functions are:

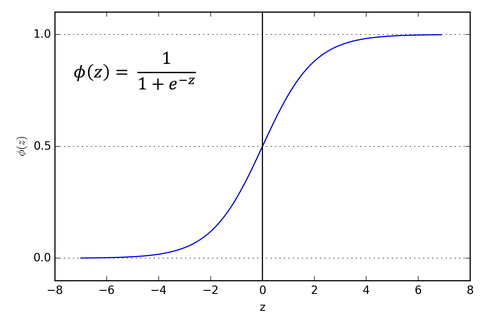
**Derivative or Differential:**Change in y-axis w.r.t. change in x-axis. It is also known as slope.

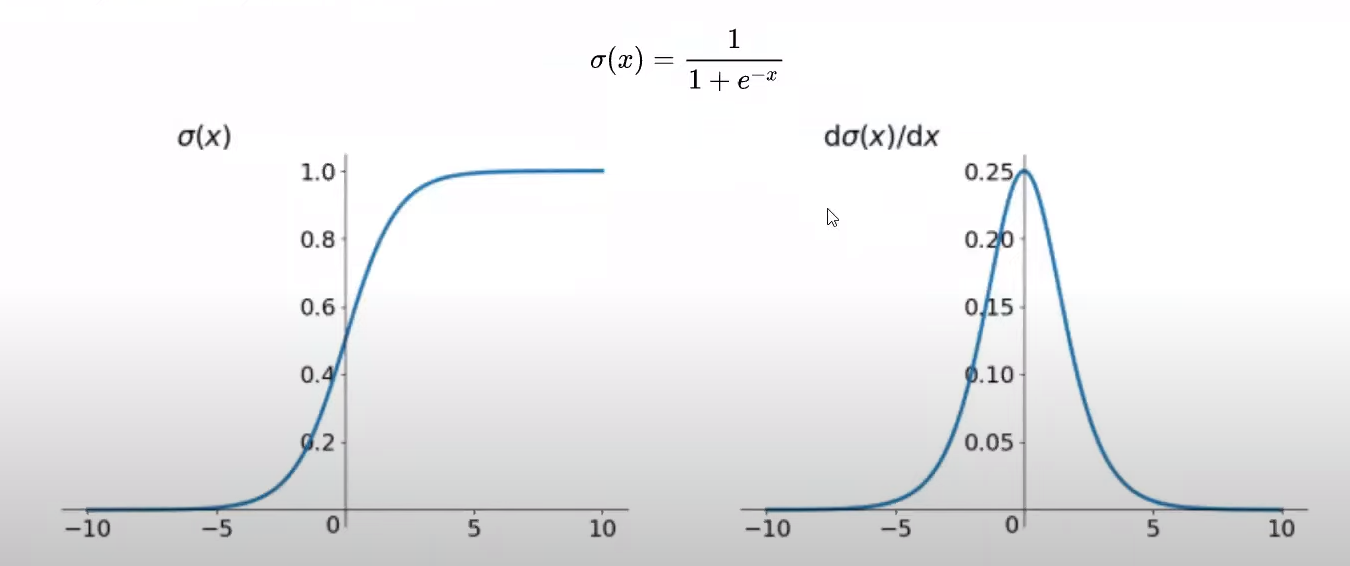
**Monotonic function:** A function which is either entirely non-increasing or non-decreasing.

The Nonlinear Activation Functions are mainly divided on the basis of their **range or curves**-

**1. Sigmoid or Logistic Activation Function (Classification problem )**

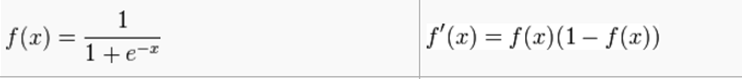
The Sigmoid Function curve looks like a S-shape.





The main reason why we use sigmoid function is because it exists between

(**0 to 1).**Therefore, it is especially used for models where we have to **predict the probability** as an output. Since probability of anything exists only between the range of **0 and 1,** sigmoid is the right choice.



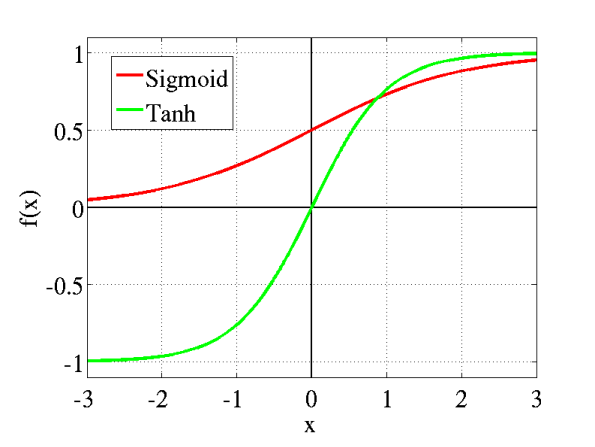
**0<=f’(x)<=0.25 value of differentiation lies like this…..**

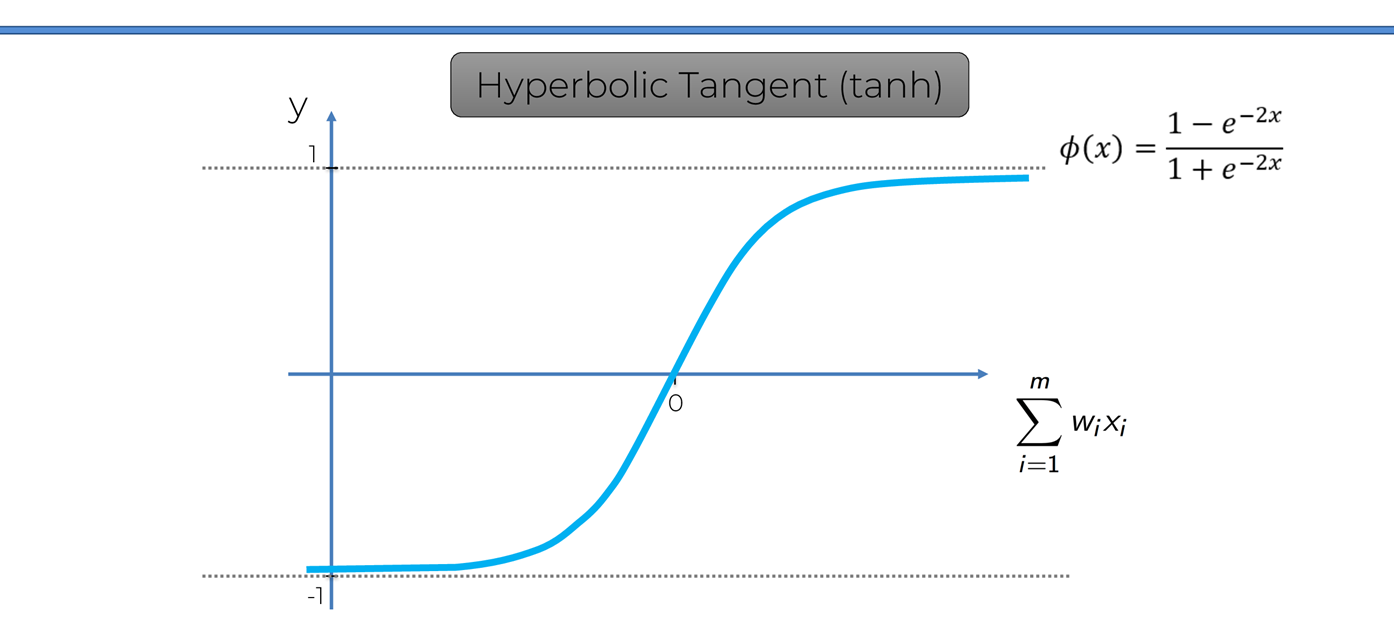
The function is **differentiable**. That means, we can find the slope of the sigmoid curve at any two points. The function is **monotonic**but function’s derivative is not. The logistic sigmoid function can cause a neural network to get stuck at the training time.The **softmax function** is a more generalized logistic activation function which is used for multiclass classification.

**Problem:-Vanishing gradient descent, Exploding gradient problem**

**2. Tanh or hyperbolic tangent Activation Function/ Threshold activation fn**

tanh is also like logistic sigmoid but **better**. The range of the tanh function is from (-1 to 1). tanh is also sigmoidal (s - shaped).

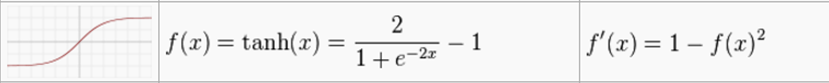




The advantage is that the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero in the tanh graph.

The function is **differentiable**.

The function is **monotonic** while its **derivative is not monotonic**.



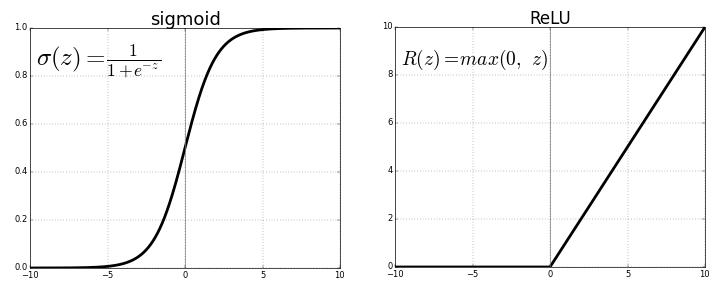
**0<=f’(x)<=1 value of differentiation lies like this…..**

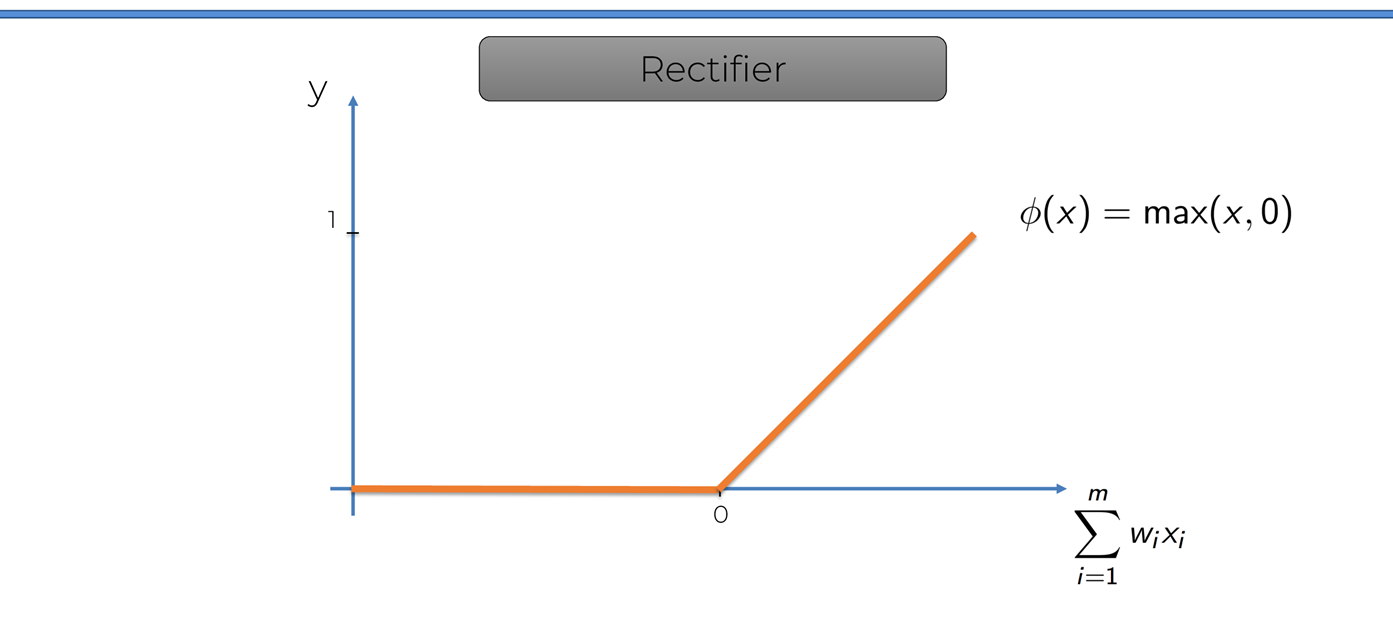
**The tanh function is mainly used classification between two classes.**

Both tanh and logistic sigmoid activation functions are used in feed-forward nets.

**3. ReLU (Rectified Linear Unit) Activation Function**

The ReLU is the most used activation function in the world right now. Since, it is used in almost all the convolutional neural networks or deep learning.





As you can see, the ReLU is half rectified (from bottom). f(z) is zero when z is less than zero and f(z) is equal to z when z is above or equal to zero.

**Range:**[ 0 to infinity)

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

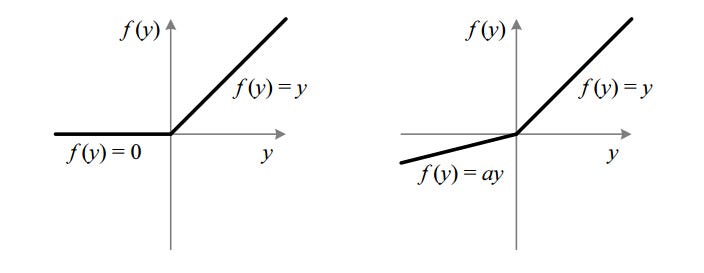


But the issue is that all the negative values become zero immediately which decreases the ability of the model to fit or train from the data properly. That means any negative input given to the ReLU activation function turns the value into zero immediately in the graph, which in turns affects the resulting graph by not mapping the negative values appropriately.

**Problem:-In back propagation in chain rule if any one derivative will come 0 then that neuron will become dead neuron . To avoid this leaky relu came .**

**4. Leaky ReLU**

It is an attempt to solve the dying ReLU problem



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

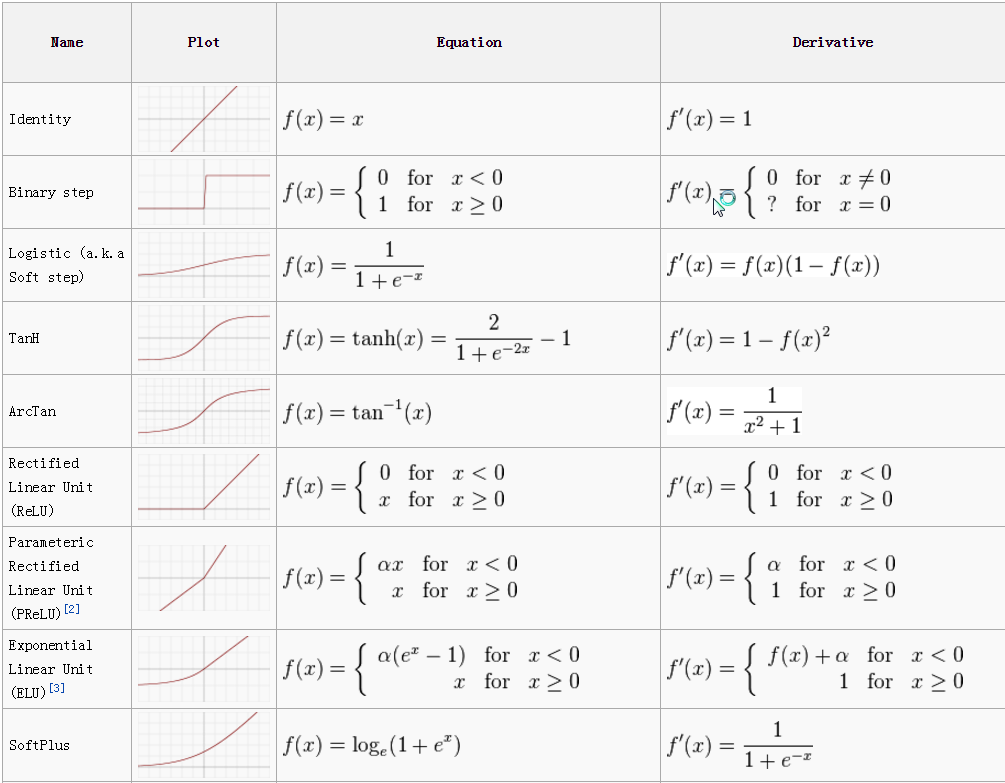
When **a is not 0.01** then it is called **Randomized ReLU**.

Therefore the **range** of the Leaky ReLU is (-infinity to infinity).



**Here a = 0.01 in leaky relu**

Other types of activation function in ipynb file.



**Fig: Activation Function Cheetsheet**

