

Activation Functions [Habiba shera]

The purpose of the activation function is to introduce non-linearity into the output of a neuron.



A neural network without an activation function is essentially just a linear regression model. So that the activation function does the non-linear transformation to the input making it capable to learn and perform more complex tasks.

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.

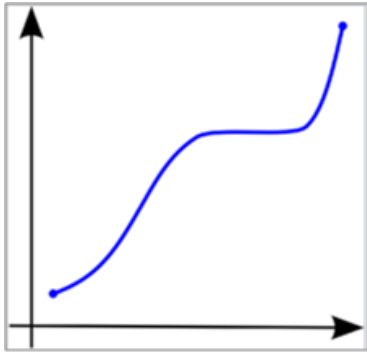


Figure 1 - A monotonically increasing function

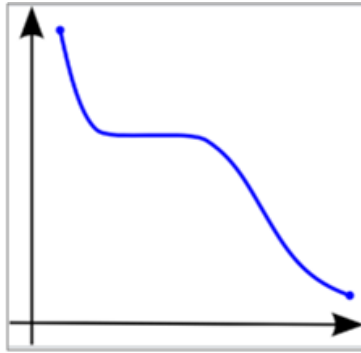


Figure 2 - A monotonically decreasing function

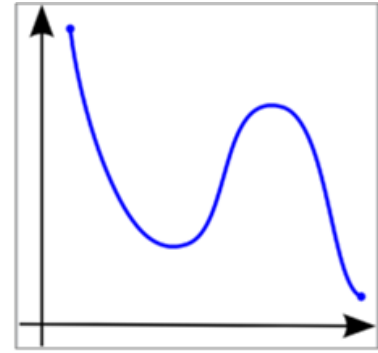
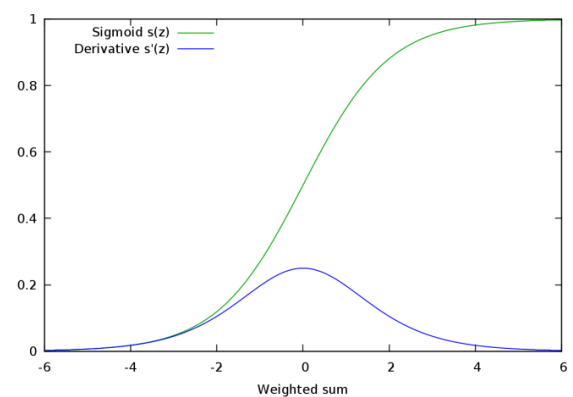
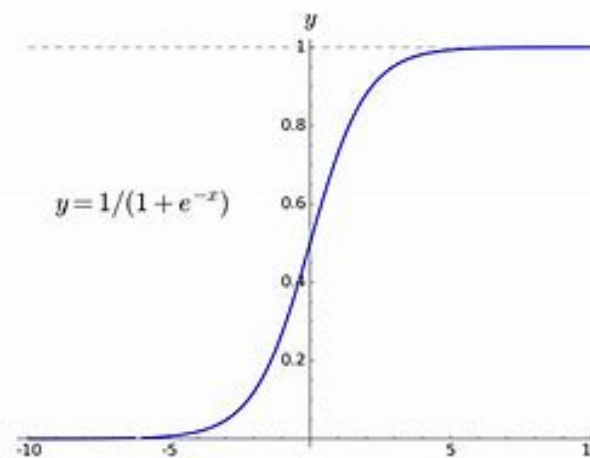


Figure 3 - A function that is not monotonic

• Sigmoid Function

- is plotted as 'S' shaped graph
- used in output layer of a binary classification
- **Value range** : {0 to 1}
- used for models where we have to **predict the probability** as an output. Since probability of anything is only between the range of **0 and 1**
- The function is **differentiable**
- The function is **monotonic** but function's derivative is not.

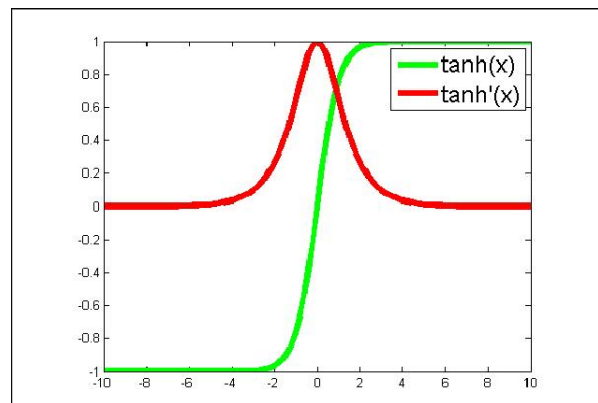
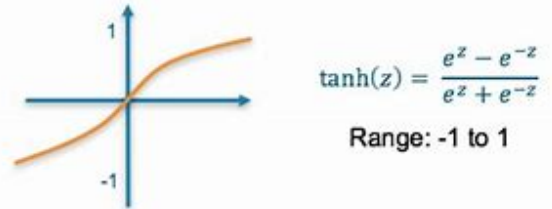


Sigmoid Function & it's derivative

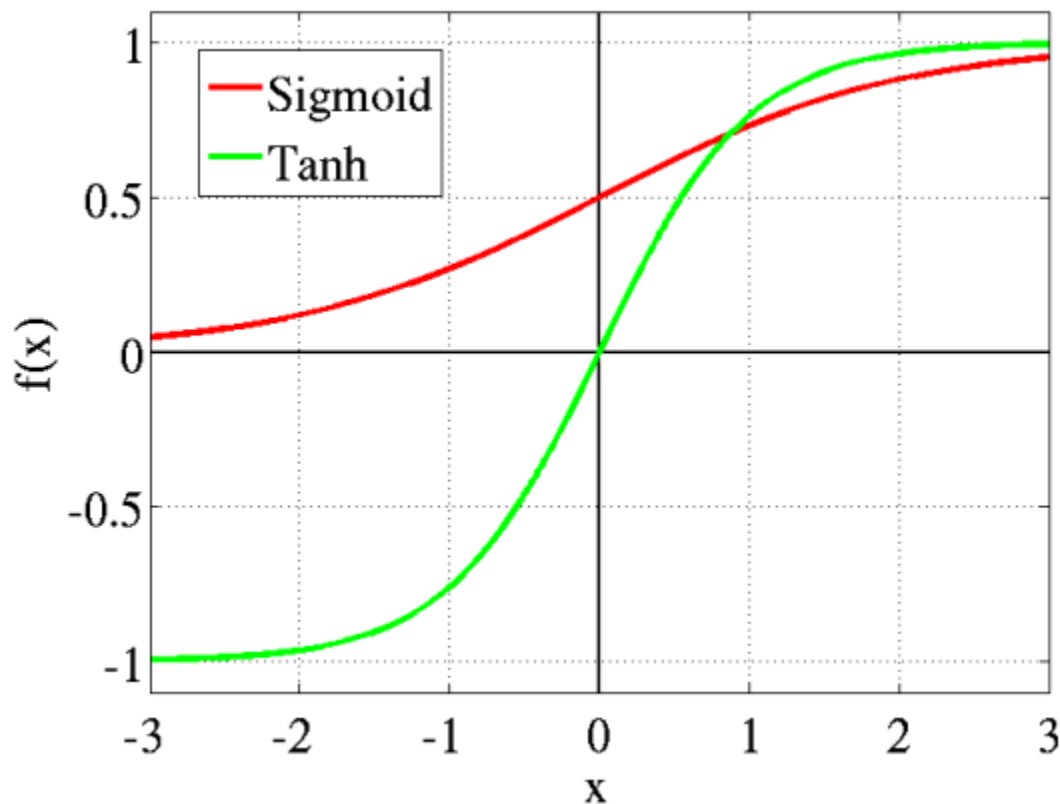
- **Tanh Function**

- always better than sigmoid function
- **Value Range : { -1 to 1 }**
- **differentiable**
- The function is **monotonic** while its **derivative is not monotonic**

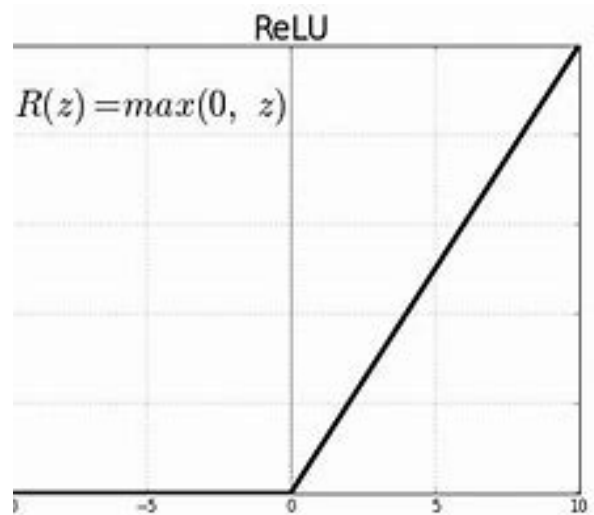
Hyperbolic Tangent Activation

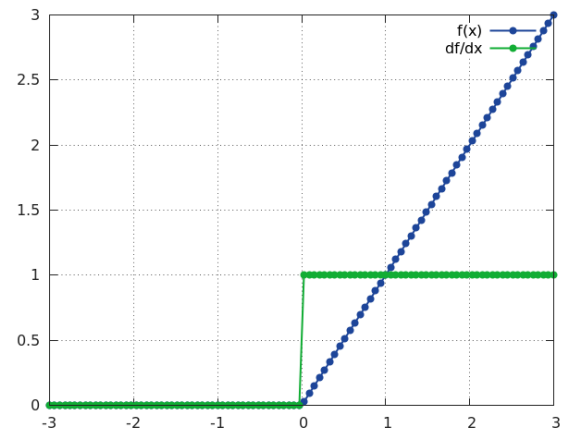


Tanh Function & it's derivative



- **ReLU (Rectified linear unit)**
 - the most widely used activation function
 - **Value Range : $[0, \infty]$**
 - The function and its derivative **both are monotonic**
 - **note1** : ReLU learns *much faster* than sigmoid and Tanh function.
 - **note2** : ReLU is less computationally expensive than tanh and sigmoid because it involves simpler mathematical operations.

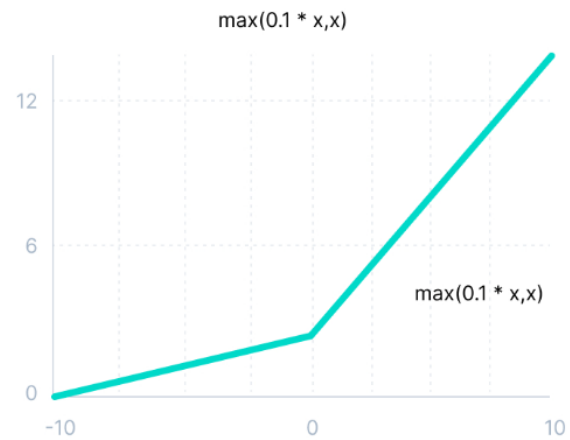


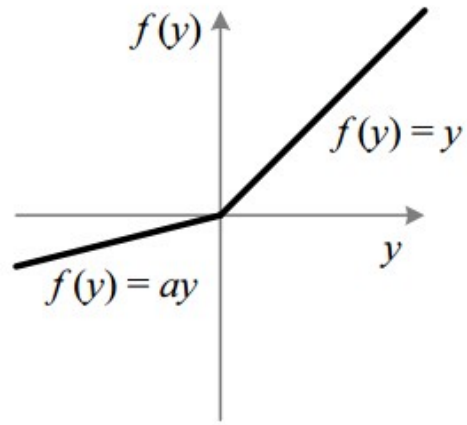
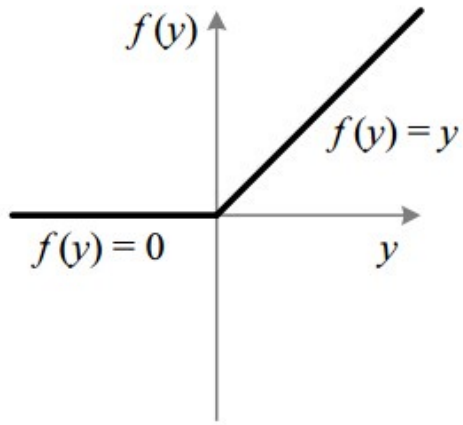


- **Leaky ReLU Function**

- improved version of ReLU function
- enable backpropagation, even for negative input values.

Leaky ReLU





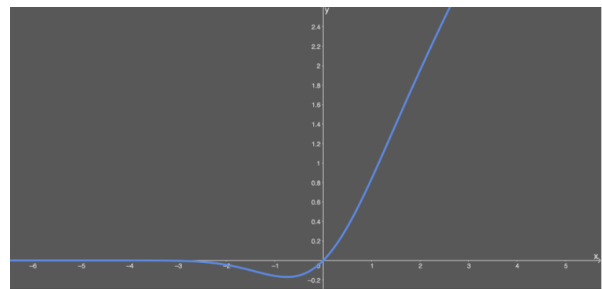
Difference between ReLU & Leaky ReLU

- **Softmax Function**

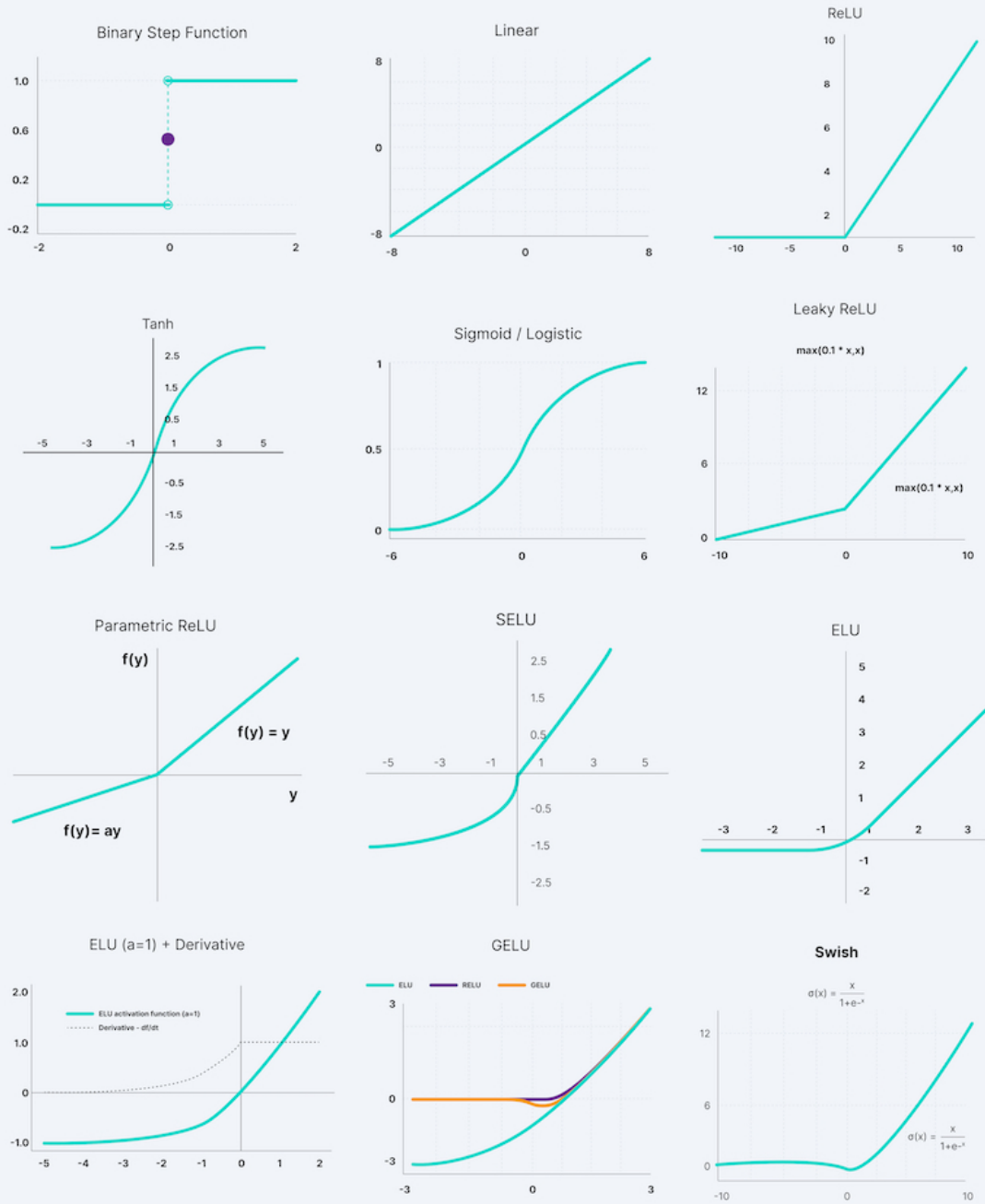
- is ideally used in the output layer of the classifier
- is described as a combination of multiple sigmoids.








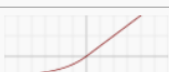

- **GELU (Gaussian Error Linear Units)**

- used in the most recent Transformers such as Google's BERT and OpenAI's GPT-2



Neural Network 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)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parametric 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		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$



How to choose the right activation function

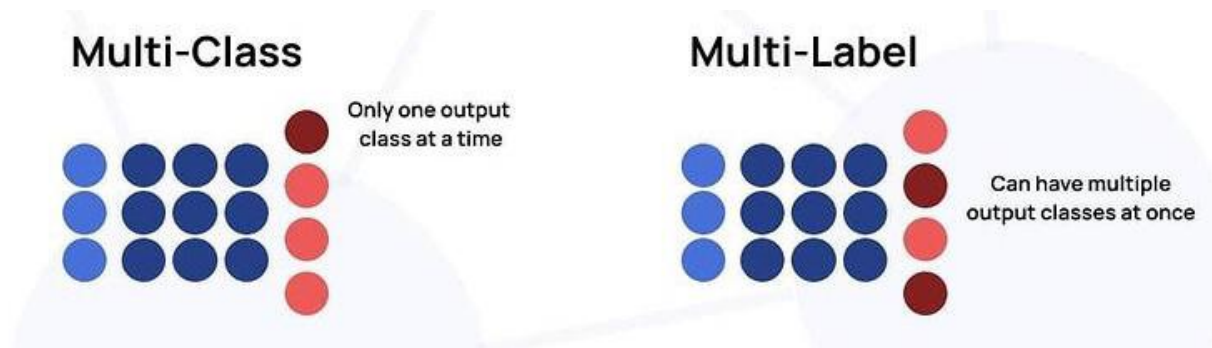
- If you don't know what activation function to use, simply use **RLUE**. “**General activation Function**”
- ReLU activation function should only be used in the hidden layers.
- If your output is binary classification, use **Sigmoid** for output layer.
- **Sigmoid** and **Tanh** functions shouldn't be used in hidden layers as they make the model more susceptible to problems during training.
- **Multilabel Classification**—Sigmoid

- **Multiclass Classification—Softmax**



In **multi-class classification** : each input will have only **one output class**.

In **multi-label classification** : each input can have **multi-output classes**.



Review	sentiment
Very good quality though	Positive
The design is very odd	Negative
I advise EVERYONE DO NOT BE FOOLED!	Negative
So Far So Good!	Positive
Works great!	positive

sentiment prediction (Is a comment is positive or negative?). Here the output should be one thing, so it's multi-class.



detect clothes (he wears hat, jeans and T-shirt)
not one thing, so it's multi-label.

- Use **Softmax** or **Sigmoid** function for the **classification** problems.
- If you have **regression problem**, use the **linear activation function** .

Feedforward Propagation *VS* Backpropagation



Feedforward Propagation : the flow of information occurs in the forward direction. The input is used to calculate some intermediate function in the hidden layer, which is then used to calculate an output. The Activation Function is a mathematical “gate” in between the input feeding the current neuron and its output going to the next layer.



Backpropagation : the weights of the network connections are repeatedly adjusted to minimize the difference between the actual output vector of the net and the desired output vector “ aims to minimize the cost function by adjusting the network’s weights and biases.”

References :

- [Activation functions in Neural Networks - GeeksforGeeks](#)
- [Activation Functions in Neural Networks \[12 Types & Use Cases\]_\(v7labs.com\)](#)
- [What are Activation Functions in Neural Networks? \(mygreatlearning.com\)](#)
- [Activation Functions in Neural Networks | by SAGAR SHARMA | Towards Data Science](#)
- [Difference between Multi-Class and Multi-Label Classification \(analyticsvidhya.com\)](#)