Activation Functions in Deep Learning: Introducing Non-Linearity

Activation functions are the unsung heroes of deep learning, adding essential non-linearity to neural networks and enabling them to learn complex patterns in data. But what exactly are they, and why are they so important?

Think of a neuron in a neural network as a tiny decision-maker. It takes in weighted inputs from other neurons, sums them up with a bias, and then applies an activation function. This function determines the neuron's "output" – whether it fires or not, and how strongly.

Here's where activation functions come in:

- They introduce non-linearity: Without them, neural networks would just be glorified linear regressions, incapable of modelling the intricate relationships in real-world data. Activation functions introduce bends and curves, allowing the network to learn complex patterns.
- They define the output range: Different activation functions have different output ranges, like between 0 and 1 or -1 and 1. This dictate how the neuron's output is interpreted, for example, as a probability in classification tasks.

Popular Activation Functions:

- Sigmoid: The classic S-shaped function, once the go-to choose, but suffers from vanishing gradients during backpropagation, limiting its use in deep networks.
- Tanh: Similar to sigmoid but cantered around zero, making it slightly better for deep networks.
- ReLU (Rectified Linear Unit): The reigning champion, ReLU simply sets negative inputs to zero, making it computationally efficient and avoiding vanishing gradients. However, it can suffer from "dying ReLU" issues.
- Leaky ReLU: A variant of ReLU that allows small negative values to leak through, alleviating the dying ReLU problem.
- Softmax: Used in the output layer for multi-class classification, it normalizes the outputs to probabilities between 0 and 1, summing to 1.

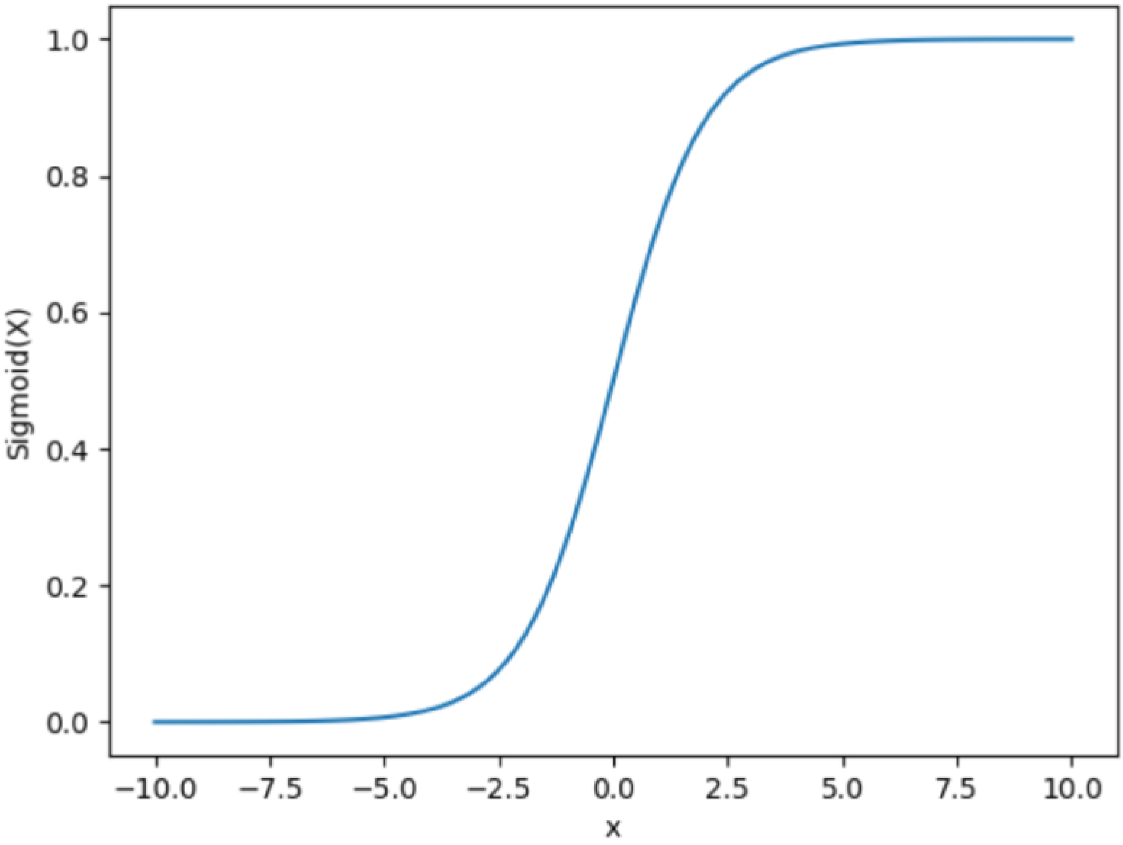
Choosing the Right Activation Function:

There's no one-size-fits-all answer. The best activation function depends on your specific task and network architecture. Here are some general tips:

- Use ReLU or Leaky ReLU for most hidden layers due to their efficiency and non-linearity.
- Use Sigmoid or Tanh if you need outputs between 0 and 1 or -1 and 1, respectively.

- Use Softmax in the output layer for multi-class classification.

Sigmoid Function



It is a function which is plotted as ‘S’ shaped graph.

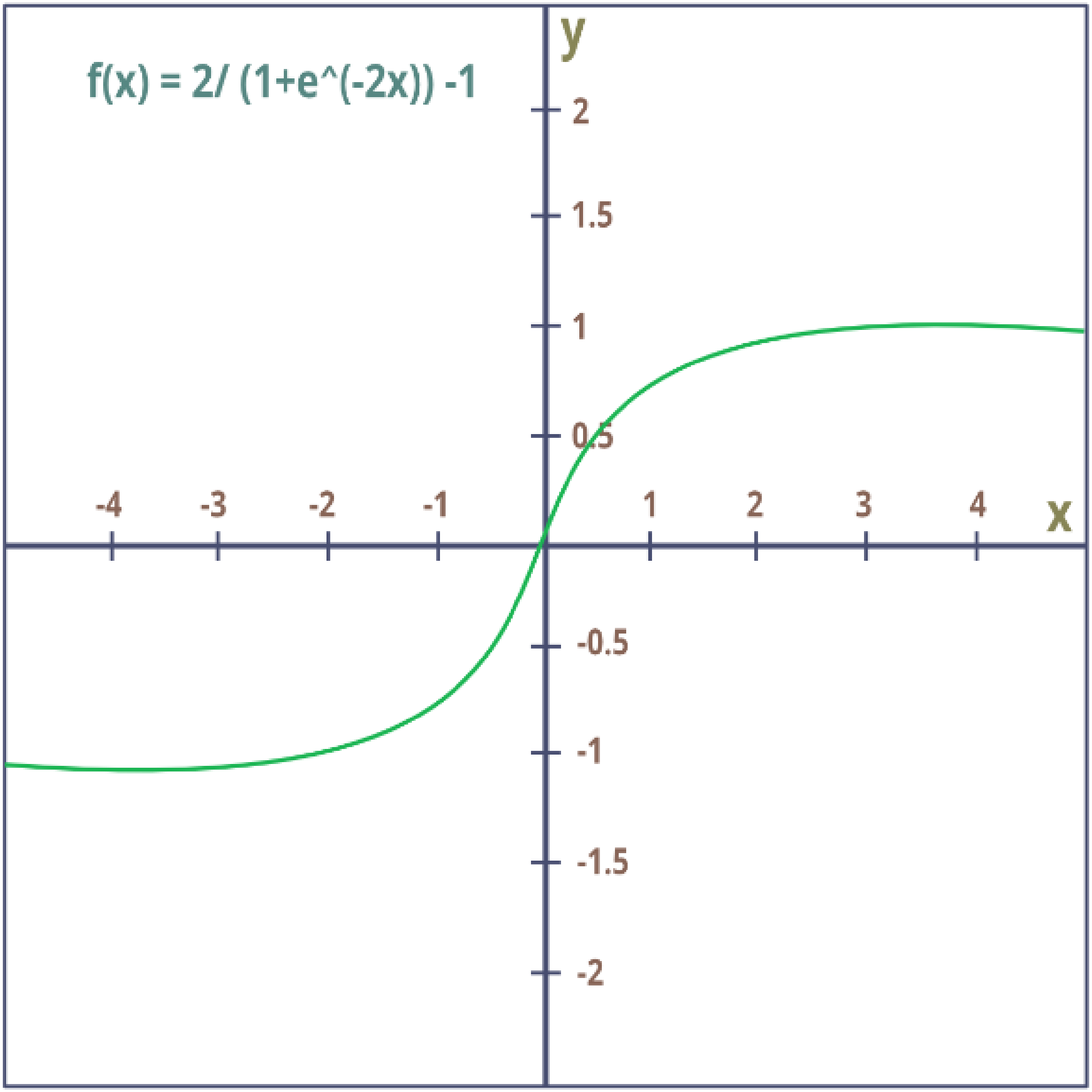
Equation : A = $1/(1 + e^{-x)}$

Nature : Non-linear. Notice that X values lies between -2 to 2, Y values are very steep. This means, small changes in x would also bring about large changes in the value of Y.

Value Range : 0 to 1

Uses : Usually used in output layer of a binary classification, where result is either 0 or 1, as value for sigmoid function lies between 0 and 1 only so, result can be predicted easily to be 1 if value is greater than 0.5 and 0 otherwise.

Tanh Function



The activation that works almost always better than sigmoid function is Tanh function also known as Tangent Hyperbolic function. It’s actually mathematically shifted version of the sigmoid function. Both are similar and can be derived from each other.

Equation :-

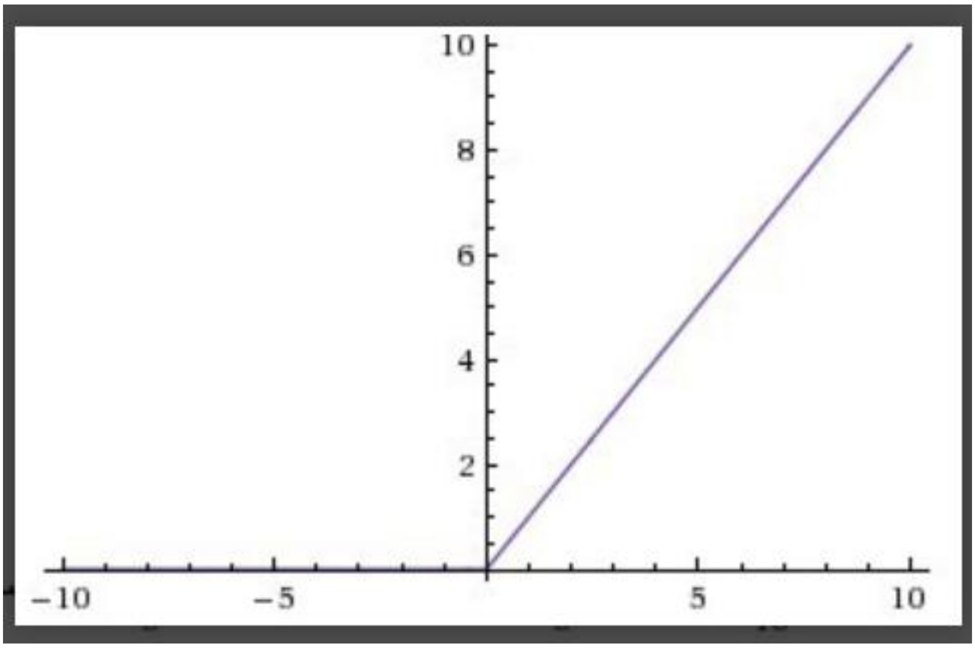
$$f(x) = tanh(x) = \frac{2}{1+e^{-2x}} - 1$$

Value Range :- -1 to +1

Nature :- non-linear

Uses :- Usually used in hidden layers of a neural network as it's values lies between -1 to 1 hence the mean for the hidden layer comes out be 0 or very close to it, hence helps in cantering the data by bringing mean close to 0. This makes learning for the next layer much easier.

RELU Function



It Stands for Rectified linear unit. It is the most widely used activation function. Chiefly implemented in hidden layers of Neural network.

Equation :- A(x) = max(0,x). It gives an output x if x is positive and 0 otherwise.

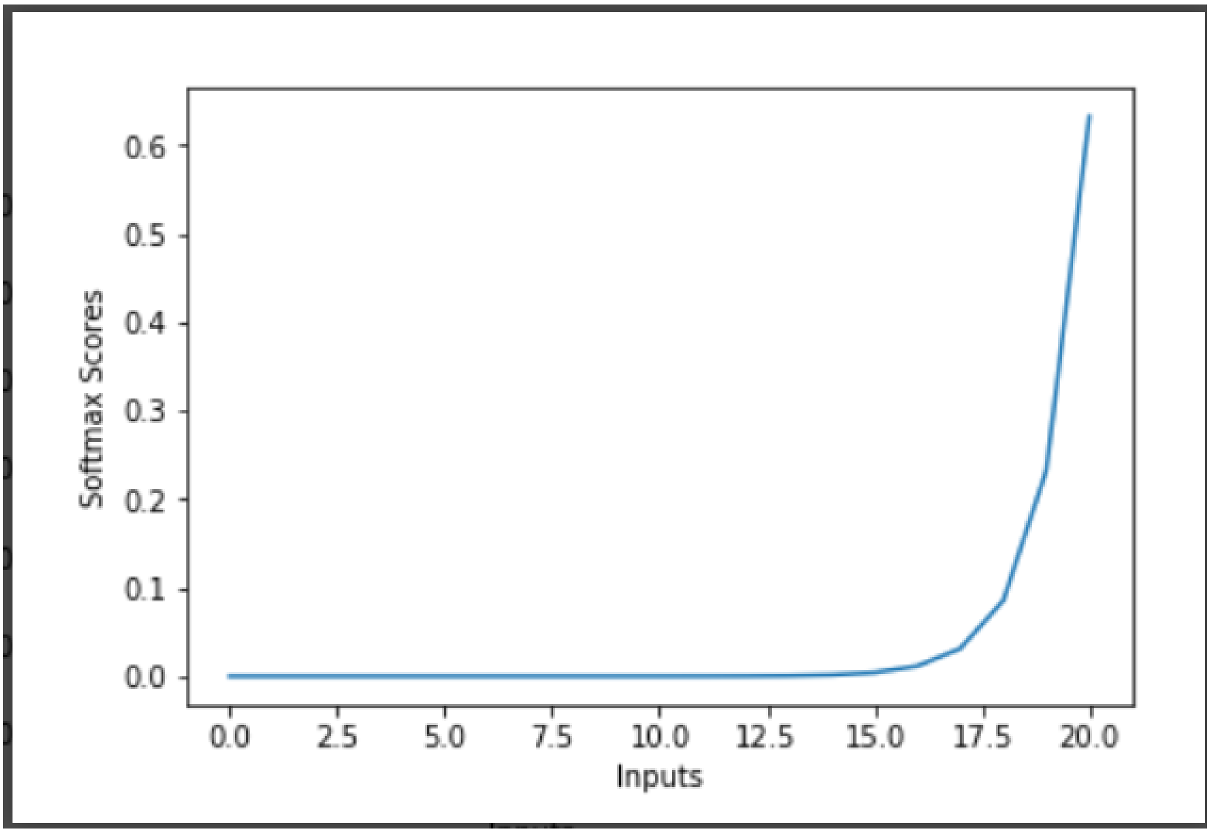
Value Range :- [0, inf)

Nature :- non-linear, which means we can easily backpropagate the errors and have multiple layers of neurons being activated by the ReLU function.

Uses :- ReLu is less computationally expensive than tanh and sigmoid because it involves simpler mathematical operations. At a time only a few neurons are activated making the network sparse making it efficient and easy for computation.

In simple words, RELU learns much faster than sigmoid and Tanh function.

Softmax Function



The Softmax function is also a type of sigmoid function but is handy when we are trying to handle multi- class classification problems.

Nature :- non-linear

Uses :- Usually used when trying to handle multiple classes. the Softmax function was commonly found in the output layer of image classification problems. The Softmax function would squeeze the outputs for each class between 0 and 1 and would also divide by the sum of the outputs.

Output:- The Softmax function is ideally used in the output layer of the classifier where we are actually trying to attain the probabilities to define the class of each input.

The basic rule of thumb is if you really don’t know what activation function to use, then simply use RELU as it is a general activation function in hidden layers and is used in most cases these days.

If your output is for binary classification then, sigmoid function is very natural choice for output layer.

If your output is for multi-class classification then, Softmax is very useful to predict the probabilities of each classes.