Activation Function

An activation function in an MLP is like a decision-maker that determines whether a neuron should contribute to the output of the network based on its input. It introduces non-linearity into the network, enabling it to learn and represent complex patterns in data.

Sometimes the activation function is called a "*transfer function*." If the output range of the activation function is limited, then it may be called a "*squashing function*."

Technically, the activation function is used within or after the internal processing of each node in the network, although networks are designed to use the same activation function for all nodes in a layer.

A network may have three types of layers: input layers that take raw input from the domain, **hidden layers** that take input from another layer and pass output to another layer, and **output layers** that make a prediction.
All hidden layers typically use the same activation function. The output layer will typically use a different activation function from the hidden layers and is dependent upon the type of prediction required by the model.

Activation functions are also typically differentiable, meaning the first-order derivative can be calculated for a given input value. This is required given that neural networks are typically trained using the backpropagation of error algorithm that requires the derivative of prediction error in order to update the weights of the model.

If Activation Function apply nahi kiye then , We would not able to capture Non-Linear Data

If Activation Function use nahi karenge then Neural Network will work & behave like a Linear Model.

Ideal Activation Function

1.) Non-Linear Activation Function (Universal Approximation Theorem)
2.) Activation Function Must be Differentiable . Why ? Becoz GD use karenge & GD Derivatives use karke Back Propagation & all krta hai
But Mandatory nahi hai E.g ReLu
3.) Computationaly Inexpensive hona chaye,
4.) Zero Centered → 1.) Zero Cen, 2.) Normalized
E.g Tanh (Speed badadeta hai)
5.) Non Saturating

What is Saturating ? E.g Sigmoid give anything it will give 0 to 1

& same with tanh it will give value from -1 to 1

This is called saturating Activation function.

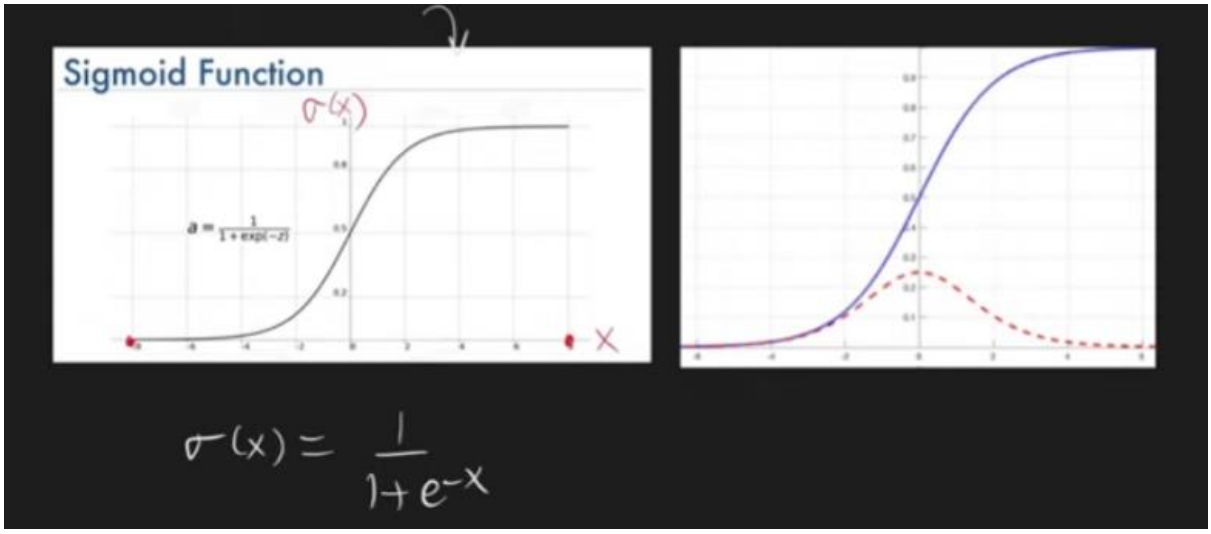
Saturating Activation Function mai Vanishing Gradient Problem aata hai

Non Saturating Function

E.g Relu max(0,x) E.g X jitna bada

All Activation Functions

Sigmoid Activation Function



Advantages of Sigmoid

1.) Sigmoid Output is between 0 to 1, we can treat it as probability E.g : Binary Classification Problem. Jitna 1 ke taraf utna yes jitna 0 ke taraf utna No.
2.) It is a Non-Linear Function

Non Linear Function mtlb it can capture Non Linearity Function

3.) It is Differentiable
In Neural Network Training back propagation we calculate Derivative
As Sigmoid is Non Linear & differentiable then easily kaam hojata sbh

Disadvantage

1.) It is saturating Function
Saturating function means esa function hai jo Dabata hai Input ko
Kindoff Squeeze krta hai.
Vanishing Gradient Problem happens Because of Saturating Function
VGD mai 0 hojata hai weights updatation so training hoti hie nahi, vaha rukjaata hai that is VGD in short

Because of This Reason we don't use Sigmoid in Hidden Layer, it is very rare, it is only used in Output layer when the problem we solving is binary classification.

2.) It is Non-Zero Centered,
Training Gets Slow.
3.) Computationaly Expensive hai becoz in Formula you can see Exponent

Mostly Sigmoid is used in Output Layer when we are doing Binary Classification.

The sigmoid activation function is a commonly used activation function in neural networks, especially in the context of binary classification problems. It squashes the input values to a range between 0 and 1, which can be interpreted as probabilities. However, the sigmoid function has both advantages and disadvantages:

Advantages:

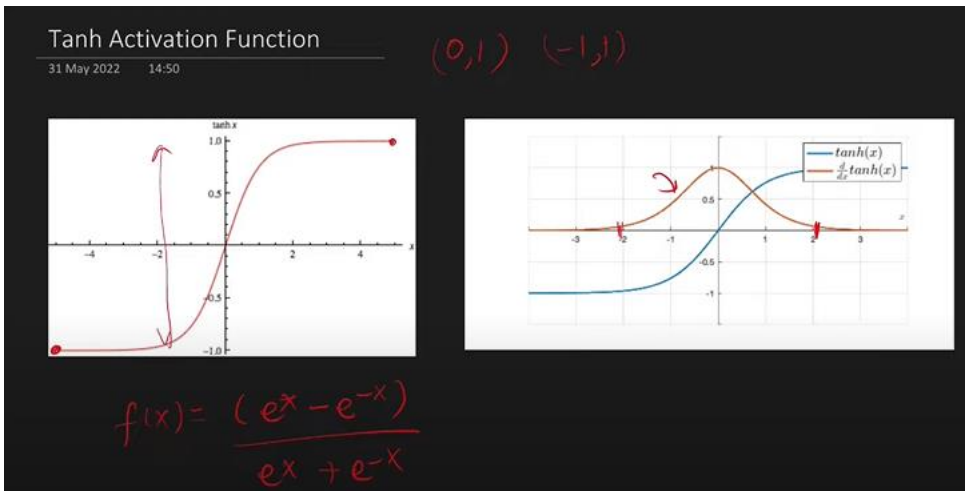
1. **Squashing Output:** The sigmoid function maps any real-valued number to the range of [0, 1]. This is useful in binary classification problems where the output can be interpreted as a probability or likelihood.
2. **Differentiation:** The sigmoid function is differentiable, which is crucial for using gradient-based optimization algorithms, such as gradient descent, in training neural networks. This allows for the calculation of gradients and facilitates the backpropagation algorithm.
3. **Historical Significance:** Sigmoid functions were historically popular and foundational in neural network literature. They have been widely used in the past, and there is a significant body of research and understanding associated with them.

Disadvantages:

1. **Vanishing Gradient Problem:** One of the major drawbacks of the sigmoid function is the vanishing gradient problem. For very large or very small inputs, the gradient of the sigmoid function becomes close to zero. During backpropagation, this can lead to very small weight updates, and the network may learn very slowly or fail to learn altogether.
2. **Not Zero-Centered:** The output of the sigmoid function is not centered around zero. This can lead to issues in weight updates during training, especially when using activation functions that produce zero-centered outputs.
3. **Saturation:** The sigmoid function saturates when the input is very large or very small, causing the output to be close to 0 or 1. This saturation can slow down the learning process because the gradients become very small, and weight updates are minimal.
4. **Output Interpretation:** While the output of the sigmoid function can be interpreted as a probability in binary classification problems, it might not be as intuitive in other contexts. For example, in multi-class classification problems, alternative activation functions like softmax are often preferred.

In practice, researchers and practitioners often use alternatives like the rectified linear unit (ReLU) and its variants due to their ability to mitigate some of the issues associated with the sigmoid function. However, the choice of activation function depends on the specific requirements and characteristics of the problem at hand.

Tanh Activation Function



It is just slightly improved version of sigmoid kindoff.

Advantages

1.) Non-Linear
2.) Differentiable
3.) It is Zero Centered
Becoz of Which The Gradients are Positive also & Negative also
So Training is Faster than Zero Centered

Disadvantages:

1.) Saturating Function, means it squeezes, isme bhi Vanishing Gradient Problem aati hai
2.) Computationally Expensive (as there is exponent)

The hyperbolic tangent function, often denoted as tanh, is another commonly used activation function in neural networks. It shares some similarities with the sigmoid function but has a different output range. Here are the advantages and disadvantages of the tanh activation function:

Advantages:

1. **Zero-Centered Output:** The tanh function produces output values in the range [-1, 1], and it is zero-centered. This property can be beneficial for optimization algorithms like gradient descent, as it helps mitigate issues like the vanishing gradient problem associated with the sigmoid function.

2. **Squashing Output:** Similar to the sigmoid function, the tanh function squashes input values to a specific range, which can be interpreted as probabilities. This is particularly useful in scenarios where the output needs to be normalized between -1 and 1.
3. **Differentiability:** The tanh function is differentiable, enabling the use of gradient-based optimization algorithms for training neural networks. This is important for efficient backpropagation and weight updates during the learning process.

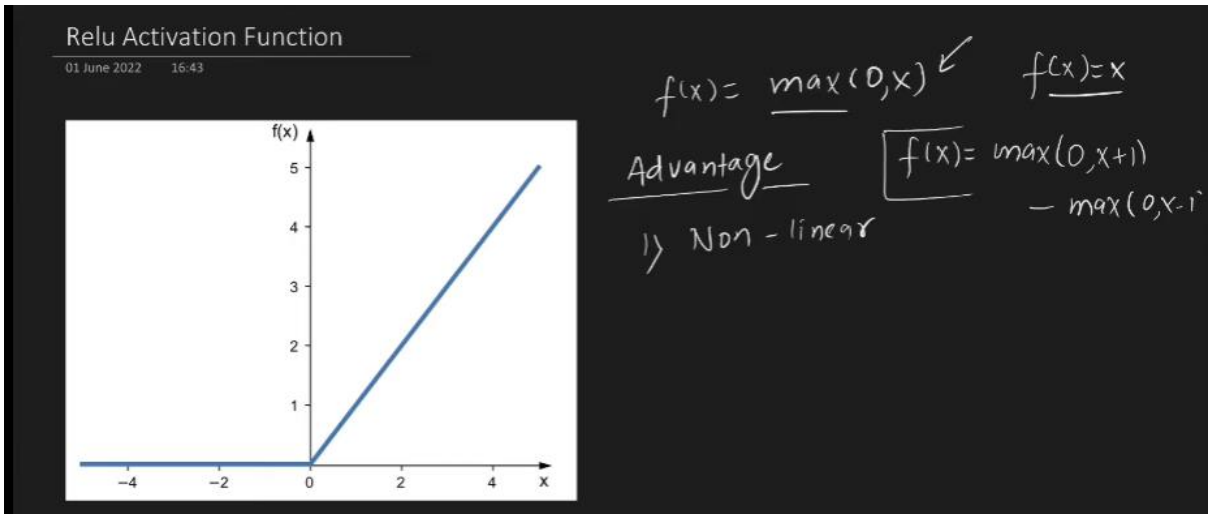
Disadvantages:

1. **Vanishing Gradient Problem:** While the tanh function helps alleviate the vanishing gradient problem compared to the sigmoid function, it can still suffer from vanishing gradients for extremely large or small input values. This can slow down the learning process, especially in deep networks.
2. **Saturation:** Like the sigmoid function, the tanh function saturates for large positive or negative inputs, leading to gradients close to zero. This saturation can affect the convergence speed during training.
3. **Output Range:** The output range of tanh is [-1, 1], which might not always be suitable for certain types of data or architectures. For some applications, other activation functions like ReLU (Rectified Linear Unit) and its variants may be preferred.
4. **Not Sparse:** The tanh function is not sparse, meaning that its outputs are always activated. In some cases, sparsity can be desirable for specific architectures or optimization goals.

In practice, the choice of activation function depends on the characteristics of the problem, the architecture of the neural network, and the specific challenges associated with training. Researchers often experiment with different activation functions to find the one that best suits the task at hand.

Relu ->

ReLU Activation Function



Advantages

1.) Non-Linear
2.) It is not saturated in Positive Region (Positive side mai kitna bhi aageh jaaskta hai)
3.) It is computationally less Expensive
4.) Convergence is Faster compared to sigmoid & tanh

Disadvantages:

1.) It is not completely Differentiable
2.) It is Not Zero Centered Just like Sigmoid

To Resolve this Problem We use Batch Normalization Technique

Batch Normalization Normalized hidden inputs

Problem of ReLu is Dying ReLu Problem

Rectified Linear Unit (ReLU) Activation Function

The Rectified Linear Unit (ReLU) is a popular activation function used in neural networks. It replaces all negative values in the input with zero, leaving positive values unchanged. Here are the advantages and disadvantages of the ReLU activation function:

Advantages:

1. **Simplicity:** ReLU is a simple and computationally efficient activation function. It only performs a thresholding operation, setting negative values to zero and leaving positive values unchanged.
2. **Mitigates Vanishing Gradient Problem:** Unlike sigmoid and tanh activation functions, ReLU does not saturate for positive input values, which helps mitigate the vanishing gradient problem. This property can lead to faster convergence during training.
3. **Sparsity:** ReLU introduces sparsity in the network since all negative values are set to zero. This sparsity can lead to more efficient computations and storage, as only a subset of neurons is activated for a given input.
4. **Promotes Non-linearity:** ReLU introduces non-linearity to the model, allowing it to learn complex patterns and representations. The model can capture more intricate relationships in the data, making it suitable for a wide range of tasks.
5. **Ease of Interpretation:** The output of ReLU is directly interpretable: if the output is positive, the neuron is activated, and if it's zero, the neuron is inactive. This can enhance the interpretability of the model.

Disadvantages:

1. **Dead Neurons:** One of the significant drawbacks of ReLU is the issue of "dead neurons." If a large gradient flows through a ReLU neuron during training, it can update the weights in such a way that the neuron will always output zero for all inputs. Once this happens, the neuron becomes inactive and doesn't contribute to the learning process.
2. **Not Zero-Centered:** The output of ReLU is not zero-centered, which can lead to issues in weight updates during training, especially when used in deeper networks or in conjunction with certain optimization algorithms.
3. **Unbounded Activation:** ReLU is unbounded on the positive side, meaning it can produce very large activations. This lack of an upper bound can lead to numerical instability and difficulties in training.

4. **Not Suitable for all Data Distributions:** ReLU may not perform well in situations where the data distribution has a significant number of negative values. This can result in a large portion of the neurons being inactive.

To address some of the issues associated with ReLU, variants like Leaky ReLU, Parametric ReLU (PReLU), and Exponential Linear Unit (ELU) have been proposed, each attempting to provide solutions to specific shortcomings while maintaining the positive aspects of ReLU. The choice of activation function depends on the specific characteristics of the problem and the data at hand.