Disha D N.

## Activation Function.

**Defination:** It is an internal state of neuron. It is used to convert the input signal of ANN node to an output signal.

→ It introduces the non linearity to the output of individual neurons or artificial neurons.

→ Activation functions decide whether a neuron should be activated or not based on input signal and weight.

→ The activation function is applied to the weighted sum of the inputs and biases of the neuron.

## Different types of activation Functions.

### ① Linear activation function.

→ It is a type of activation function, which outputs a linear combination of input values.

→ It is the simplest activation function and is known as the identity function.

Formula: $f(x) = x$

→ Here, the input value $x$ is the weighted sum of the inputs and bias of the neuron, and output of the function is also the same as input value.

→ The linear activation function has a constant slope and doesnot introduce any non-linearity to the neural network's output.

→ Hence it unable to learn complex relationship betw input features

→ Linear activation functions are typically used in regression problems

## ② sigmoid activation

→ It is a type of activation function which maps the input to a value between 0 and 1.

→ It is most commonly used activation function in binary classification problems

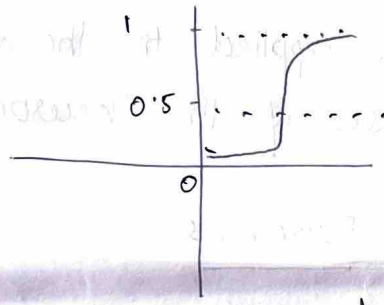


Fig: Sigmoid activation function

Mathematically,

$$f(x) = \frac{1}{1 + e^{-x}} \longrightarrow (*)$$

Here the input value x is the weighted sum of the inputs and biases of the neuron, and output of the function is a value between 0 and 1

Eg:

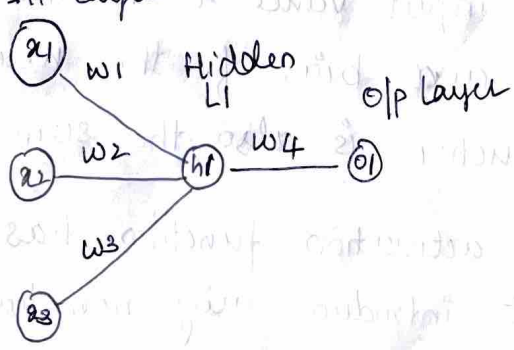


fig 2: simple neural network

③ ReLU activation Function.

→ ReLU (Rectified Linear Unit) is a widely used activation function in neural networks.

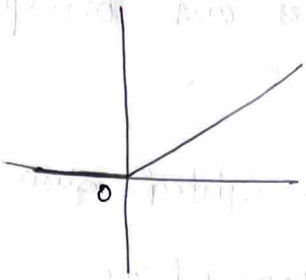


Fig: ReLU activation function graph.

Mathematical formula:

$$f(x) = \max(0, x)$$

→ In other words, the ReLU function returns 0 if the input is negative, else it returns input value

→ The main advantage of using the ReLU activation function is that it is computationally efficient and allows for faster training.

→ This is because the derivative of ReLU is either 0 or 1, which makes it easy to compute during back propagation.

→ It helps to address the vanishing gradient problem, which is common issue in deep neural networks

→ Problem: Dead activation function

④ Tanh activation function. [Hyperbolic Tangent]

→ Tanh activation function maps the input value to a value between -1 and +1

→ Mathematical formula for tanh activation function

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

where e = Euler's number
e ≈ 2.718

→ Consider a neural network as in fig 2, during feed forward process, 2 steps will be performed at each neuron.

→ Hidden layer1 has one hidden neuron h1, Input layer has 3 features and corresponding weights are applied

step1: Calculation of weighted sum and bias

$$y = (x_1 w_1 + x_2 w_2 + x_3 w_3) + \text{bias}$$

$$y = \sum_{i=1}^{3} x_i w_i + b \rightarrow (1)$$

step2: Apply activation function ie sigmoid in this example to equation (1)

$$Z = \text{Activation}(y)$$

ie Applying the activation formula from equation(*)

$$Z = \frac{1}{1 + e^{-y}}$$ where y is from equation (1), we get the value of Z which is in the range of 0 and 1.

Points to Remember for sigmoid activation function

* Sigmoid activation function has a derivative that is easy to calculate
* Major drawback is when the input value becomes very large or very small, the gradient of the function approaches zero.
* It leads to vanishing gradient problem, which can make training difficult for deep neural network

→ The tanh function is a sigmoid function, which means that it has S-shaped curve.

→ It is similar to the logistic sigmoid function, which maps the input to a value between 0 and 1, but tanh maps the value between -1 and +1

→ The main advantage of using the tanh function is that it is a smooth, differentiable function that can be used for both forward propagation and backpropaga-tion.

→ The tanh function is a symmetric function, which means that is centered around 0. Hence it can be useful in certain types of neural network architectures such as auto-encoders, where input and output layers are symmetric.

→ The major limitation is vanishing gradient, which can occur when the gradients become very small, making it difficult for the network to learn.

(5) Leaky ReLU:

→ It is the variation of standard ReLU activation function that addresses some of the limitations of the ReLU function such as "dying ReLU" problem.

Formula, $f(x) = \max(ax, x)$

where $x$ is the input value to the function and $a$ is a small constant, usually set to 0.01. The function returns the input value if it is positive, and returns the product of the constant and the input value if it is negative

→ The main advantage of using leaky ReLU is that it allows a small, non-zero gradient to be propagated through the network when the input is negative.

→ Another advantage of using the Leaky ReLU function is that it can help to reduce the likelihood of overfitting in the network. This is because the function introduces some amount of noise into the output, which can help to prevent the network from memorizing the training data too closely.

→ one ~~disto~~ disadvantage of using the Leaky ReLU function is that it can be slower to compute than the standard ReLU function.