Step ➔

Step activation is a binary function that returns 1 if the input is larger than or equal to the threshold value, and 0 otherwise. It is effective for binary classification problems and can be used in neural networks as an activation function. Yet, its discontinuous nature makes gradient-based optimization algorithms such as backpropagation challenging to implement. The threshold value must be selected with care, as it influences the output of the function. The step function is straightforward and easily interpretable, but its output is not smooth, making it less ideal for regression situations. Also known as the Heaviside step function and the unit step function. f(x) = 0 if x 0 and f(x) = 1 if x > 0 is the definition of the step function. The step function is advantageous for issues requiring a binary output due to its simplicity and interpretability. However, its discontinuous nature restricts its application to more intricate issues

Sigmoid➔

The sigmoid activation function is a mathematical function that transfers every input value to a value between 0 and 1, making it helpful for neural network and logistic regression models. It operates by applying the sigmoid function to the weighted sum of inputs, generating a value between 0 and 1 that can be read as the probability of an input belonging to a particular category. However, the sigmoid function has some shortcomings, including the vanishing gradient problem, saturation at the extremities of its range, not being zero-centered, and needing the costly computation of exponentials. These limitations have prompted the creation of more successful alternatives, such as the ReLU and its derivatives, making the sigmoid function less prevalent in contemporary neural networks. Despite these restrictions, the sigmoid function is nonetheless straightforward to comprehend and interpret.

Tanh➔

Tanh is an activation function used in neural networks that maps input values to output values between -1 and 1. It works by taking the exponential function of the input, subtracting the exponential function of the negative input, and dividing the difference by the sum of the exponential function of the input and the exponential function of the negative input. Tanh is non-linear, making it capable of modeling complex relationships between input and output variables. It has a derivative that makes it useful in gradient-based optimization algorithms. However, Tanh is also prone to the vanishing gradient problem and saturation at the extremes of its range, which can slow down learning. The computation of exponentials required for Tanh can also be computationally expensive. Overall, Tanh is a useful activation function in certain types of neural networks, but it has limitations and drawbacks that should be considered when selecting an appropriate activation function.

Relu➔

The activation function in a neural network is responsible for changing the sum of the weighted input from the node into the node's activation or output for that input.The rectified linear activation function, or ReLU, is a piecewise linear function that outputs the input directly if it is positive and 0 otherwise. It has become the default activation function for a variety of neural network types because models that employ it are easier to train and frequently produce superior performance.the rectified linear activation function for deep learning neural networks

Due of the vanishing gradient problem, the sigmoid and hyperbolic tangent activation functions cannot be employed in networks with numerous layers.The rectified linear activation function solves the problem of vanishing gradients, enabling models to learn more quickly and perform better.For creating multilayer Perceptron and convolutional neural networks, the rectified linear activation is the default activation.

Elu➔

The Exponential Linear Unit (ELU) is a neural network activation function used to construct neural networks that can learn complex input-output interactions. It is a popular choice in deep learning due to its ability to counteract the vanishing gradient problem, boost performance, and resist noise. The ELU function applies an exponential function to x, subtracts 1 and multiplies by alpha when x is negative, resulting in a smooth shift from negative to positive numbers. The alpha parameter governs the slope of this curve; normally, a small value is used such that the function is almost linear for tiny negative values. The Exponential Linear Unit (ELU) activation function offers various advantages over other activation functions, such as the ReLU activation function, such as enhanced performance, a reduced gradient vanishing problem, and increased noise resistance. However, it is computationally expensive and difficult to implement in certain hardware or software environments, and may not be appropriate for certain tasks, such as picture segmentation.

Selu➔

The SELU (scaled exponential linear unit) activation function is a form of activation function used in artificial neural networks. It is used to automatically equalize the outputs of each layer in a neural network, which can help eliminate the vanishing gradient problem and increase the efficiency of training. SELU is formally represented as f(x) = lambda * x if x > 0 lambda * alpha* x if x 0. (exp(x) - 1) if x <= 0where alpha and scale are constants computed from the mean and standard deviation of the input data. Self-normalization is one of the benefits of utilizing the

SELU activation function, as it improves the performance of deep neural networks, especially those with numerous layers.

However, initialization of network weights is essential for the effective operation of SELU and can be a difficult process. Additionally, SELU may not be as effective for networks with particular architectural or data sets.