|  |
| --- |
| Activation  Function |
|  |
| Assignment 2  Computer Vision and Pattern Recognition  Section : C  Submitted by :  Pritom Debnath(20-42414-1)  Submitted to : Debajyoti Karmaker |



# Activation Functions

|  |
| --- |
| Step Activation Function: A basic non-linear activation function is the step activation function. It accepts an input value and provides a binary output of 0 or 1, depending on whether the input value is greater than or equal to a predefined threshold.  The equation for this function is- |
| Figure 1: Step function |
| The result is a horizontal line at 0 for input values less than or equal to the threshold, and a horizontal line at 1 for input values greater than or equal to the threshold, as shown in the graph. It only returns 1 when the input value is larger than the threshold. As a result, it is also known as the threshold activation function. It is frequently used in binary classification issues, such as spam identification or sentiment analysis, where the result must be either 0 or 1. Yet, because of its discontinuous character, it might present issues with gradient-based optimization techniques in particular instances.  Sigmoid Activation Function:  The sigmoid function is a useful activation function that performs non-linear changes of input values, particularly in the context of probabilistic output, which implies that the function's output is always between 0 and 1. When the input value grows or decreases, the graph resembles a "S" shaped curve that gradually approaches an upper and lower limit.  The equation for the sigmoid function is:   |  | | --- | | Figure 2: Sigmoid function |   Tanh Activation Function:  Tanh activation functions are sigmoid functions with values ranging from -1 to 1. Its graph is symmetric about the origin and has a "S" form (0,0).  The formula of tanh function is:   |  | | --- | | Figure 3: TanH function | |

ReLu Activation Function:

The Rectified Linear Unit (ReLU) activation function provides a graph that resembles a "hinge," with the output being 0 for all negative inputs and growing linearly for positive inputs.

The formula for this function is:

Or,

|  |
| --- |
| Figure 4: ReLu |

For all positive input values, the graph of the ReLU function is a straight line with a slope of one, and for all negative input values, it is a flat line with an output of zero. The graph shows a dramatic bend at the origin where the function changes from flat to straight. It is commonly utilized due to its efficiency, sparsity, and strong generalization performance. One possible disadvantage of this function is that any negative input supplied to the ReLU activation function immediately converts the value to zero in the graph, affecting the resultant graph by not mapping the negative values effectively.

SeLu Activation Function:

The Scaled Exponential Linear Unit (SELU) activation functions that induce self-normalizing properties.

The formula for this function is-

|  |
| --- |
| Figure 5: SeLu |

Here alpha(α) ≈ 1.67326324 and scale (λ) ≈ 1.05070098

If x is a positive number, the output is x\*λ; if x is 0 or a negative value, the result is a function that goes up to 0. One of its benefits is that it can self-normalize the hidden units, ensuring that the output of each layer has a zero mean and unit variance. This can result in faster convergence and improved performance, particularly in deep neural networks. This function has the potential to outperform ReLu. Nevertheless, because to its complexity and instability, it is not appropriate for all types of neural networks and data.

ELU Activation Function:

ELU is an activation function that has a quicker cost-to-zero convergence and produces more accurate results.

The formula for this function is-

|  |
| --- |
| Figure 6: Elu |

In this case, alpha is a hyperparameter that defines the function's value when x is negative.

The eLu function's graph is similar to the ReLU function's, but it has a smooth curve for negative values of x, which can help address some of the ReLU function's drawbacks, such as the "dying ReLU" problem, which occurs when a large number of neurons in a network give zero output and are unable to learn. When x is positive, the eLu function works similarly to an identity function, returning x, but it can also destroy the activation with an output range of [0, inf], which is a disadvantage of this function. When x is negative, the function returns a smoothed version of the exponential function that is scaled by alpha. This enables the function to have a non-zero gradient for negative x values, which can aid in the prevention of the "dying ReLU" problem.