Activation Functions

The Step function is a binary function that outputs a value of 1 if the input is greater than or equal to a threshold value and 0 otherwise. On the other hand, the Sigmoid function is a smooth function that maps any input to a value between 0 and 1. The step function has a discontinuous graph and is not differentiable at the threshold point. However, the sigmoid function has a continuous and differentiable graph, making it suitable for optimization algorithms such as gradient descent.

The Step function is mainly used in binary classification problems, while the Sigmoid function is used in logistic regression and neural network models. After this short comparison, we can see that the sigmoid function is better than the step function. However, there are many conditions where the step function will be the best choice against the sigmoid function.

Sigmoid and Tanh functions share similarities but differ in range, shape, symmetry, and applicability. For example, the sigmoid function maps any input to a value between 0 and 1. At the same time, the Tanh function maps any input to a value between -1 and 1. The Tanh function has a "tanh"-shaped curve that is symmetrical around the origin, whereas the sigmoid function has an "S"-shaped curve. In the vicinity of the origin, the Tanh function exhibits steeper gradients than the sigmoid function. The sigmoid function is not symmetric around the origin, while the Tanh function is symmetric. This property of the Tanh function allows for negative and positive inputs.

ReLU (rectified linear unit) activation functions are continuous activation functions that return either 0 or the input value depending on whether the input is greater than or less than 0. This type of activation function is most commonly used in deep learning networks. As the ReLU function is non-linear, it is simple to backpropagate errors and use it to activate neurons across many layers. Also, this function accelerates the convergence of stochastic gradient descent compared to the sigmoid and tanh activations. It is computationally efficient, allowing the network to converge very quickly. It is capable of outputting an actual zero value. ReLU output is not zero centered; it decreases the efficiency of the neural network. During backpropagation, the weights' gradients will be either uniformly positive or negative. A dying ReLU problem occurs when the neuron gets stuck on the negative side and constantly outputs zero. This arises when the learning rate is high or when the negative bias is significant.

ELU and SELU functions have a similar shape for inputs greater than zero, but SELU has a specific shape for inputs less than or equal to zero, a scaled exponential function. Moreover, the common thing between them is that both functions are continuous and differentiable at all points. In the vanishing gradient issue, SELU has been shown to perform better than ELU in avoiding the problem. The normalization term in the SELU function indicates that the function's output is normalized with the input distribution, leading to quicker convergence and improved generalization. SELU function requires additional normalization terms, whereas ELU only requires the computation of exponential functions.

Choosing the appropriate activation function based on the specific task and dataset is essential. The choice of activation function depends on the problem at hand, and it is essential to select the proper activation function to ensure better performance and faster convergence.