Step function: A step function is a function that changes value suddenly at specific points. A step function is defined as a function that takes a real number as input and outputs a constant value for a specific range of inputs. The simplest example of a step function is the Heaviside step function, also called the unit step function.

Advantage:

- The step function is a straightforward mathematical operation that is not only easy to comprehend but also straightforward to put into practice. It is a piecewise constant function, which indicates that it maintains a constant value throughout a certain range of inputs, and this is the definition of a constant function.
- When dealing with binary classification issues, where the output of a neural network must be either 0 or 1, the step function is an effective tool to utilize. It is possible to use it as an activation function in order to convert the value produced by a neuron into a binary representation.
- The step function is computationally efficient, and it is easy to calculate its output for any given input.

Disadvantage:

- The step function is not a differentiable function its derivative does not exist at the points where it is applied. Because of this, it is difficult to use in certain machine learning algorithms, such as gradient descent, that require the calculation of derivatives.
- The step function is not suitable for problems where the output needs to be a continuous value. Since it is a piecewise constant function, it can only produce a limited number of output values.
- Because of the gradient vanishing issue, training deep neural networks may be challenging when using the step function. The training process may be slowed considerably since the derivative of the step function is zero for all inputs except the step points.

Sigmoid function: The sigmoid function is a type of activation function that is commonly used in artificial neural networks. It is a mathematical function that maps any input value to a value between 0 and 1. The sigmoid function is defined as: $\sigma(x) = \frac{1}{1+e^{(-x)}}$

The sigmoid function is a smooth, S-shaped curve that is useful for problems where the output needs to be a probability between 0 and 1, such as binary classification problems. The output of the sigmoid function can be interpreted as the probability of the input belonging to a certain class.

Advantage:

- The output of the sigmoid function is bounded between 0 and 1, which makes it useful for problems where the output needs to be interpreted as a probability. For example, in binary classification problems, the output of a sigmoid function can be interpreted as the probability of the input belonging to a certain class.
- The sigmoid function is a differentiable function, which means that its derivative can be calculated at any point. This property is essential in many machine learning algorithms that use gradient-based optimization methods, such as backpropagation, to update the weights of a neural network during training.

Disadvantage:

- The sigmoid function can suffer from the problem of vanishing gradients, especially when used in deep neural networks. This is because the gradient of the sigmoid function becomes very small for large values of the input, which can make it difficult to update the weights of the network during training.
- The computation of the sigmoid function can be computationally expensive, especially when dealing with large datasets. This is because the function involves the computation of the exponential function, which can be time-consuming.

TanH: The hyperbolic tangent (tanh) activation function is a type of mathematical function commonly used in artificial neural networks. The tanh function has a range between -1 and 1, meaning that it squashes its input values into this range. This can be useful in neural networks as it allows the outputs of the network to be normalized and centered around zero, which can help with training.

The formula for the tanh function is tanh(x) = $\frac{e^{2x}-1}{e^{2x}+1}$

The output of the function will always be a value between -1 and 1, regardless of the input. The tanh function is similar to the sigmoid function, but with a steeper slope in its central region. This means that the tanh function is often preferred over the sigmoid function for activation in neural networks, as it can provide a stronger signal and better performance.

Advantage:

- The tanh function is a non-linear activation function, which allows neural networks to model complex, non-linear relationships between inputs and outputs.
- The tanh function is symmetric around zero, which means that its output values are centered around zero. This can help with training, as it can prevent the vanishing gradient problem that can occur when using some other activation functions.
- The tanh function is differentiable, which allows for gradient-based optimization methods to be used during training.

Disadvantage:

- The tanh function can suffer from vanishing gradients. This means that the gradients can become very small as the input values become very large or very small, making it difficult to train the neural network.
- The tanh function involves calculating exponential functions, which can be computationally expensive. This can slow down the training process, particularly when dealing with large neural networks.
- The output of the tanh function can saturate to the extremes (-1 or 1) for large input values, which can limit the representational power of the neural network.

ReLU: It is a simple, yet effective, non-linear activation function that transforms an input signal to an output signal by outputting the input if it is positive, and outputting zero if it is negative.

The formula for the ReLU function is f(x) = max (0, x)

where x is the input value to the function, and f(x) is the output value. The ReLU function outputs the input value if it is positive (greater than zero), and outputs zero if it is negative (less than or equal to zero). The ReLU function has become a popular choice for activation in deep neural networks, due to its simplicity and effectiveness.

Advantages

- The ReLU function is a simple function that can be computed very quickly, which makes it ideal for large neural networks.
- Like other activation functions, the ReLU function is non-linear, which allows neural networks to model complex, non-linear relationships between inputs and outputs.
- The ReLU function avoids the vanishing gradient problem that can occur with other activation functions, such as the sigmoid and tanh functions. This is because the gradient of the ReLU function is either 1 (for positive inputs) or 0 (for negative inputs), which prevents the gradients from becoming too small during training.

Disadvantage:

- The ReLU function can suffer from the "dying ReLU" problem, where neurons that have a negative output value always output zero, and therefore do not contribute to the network's output. This can lead to a decrease in the network's representational power and can make it difficult to train the network.
- The ReLU function is unbounded, which means that the output values can become very large (i.e., positive infinity) for very large input values. This can lead to numerical instability and difficulty during training.

ELU: The Exponential Linear Unit (ELU) is an activation function commonly used in artificial neural networks. The ELU activation function is a differentiable function that maps the input values to an output range of (-1, ∞).

The ELU activation function is defined by the following equation:

$$ELU(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x < 0 \end{cases}$$

where x is the input to the function, and α is a hyperparameter that determines the value that the function approaches for large negative values of x. The parameter α is usually set to 1.0.

Advantage:

- The ELU function can produce negative values, unlike ReLU, which only outputs 0 for negative inputs. This helps prevent the "dead neuron" problem, where a neuron in the network may stop learning if it constantly receives only negative inputs.
- The ELU function has a mean output closer to zero compared to ReLU, which has a mean output of 0.5. This reduces the bias shift, which can help speed up learning and improve model performance.
- The ELU function has a smooth derivative, which can help avoid problems like the "exploding gradient" problem that can occur with activation functions like sigmoid and tanh.

Disadvantage:

- The ELU function involves computing exponentials, which can make it slower to compute compared to other activation functions like ReLU or sigmoid.
- The ELU function has a hyperparameter (**alpha**) that controls the slope of the function for negative inputs. This parameter needs to be tuned during training, which can require extra computational resources and time.
- The exponential function used in the ELU function can potentially lead to numerical instability if the input is very large or very small, which can cause problems during training.

SELU: The SELU activation function is a variant of the Exponential Linear Unit (ELU) activation function that has been shown to work well in deep neural networks.

The SELU activation function is defined by the following equation:

$$SELU(x) = \lambda\begin{cases} x & \text{if } x \geq 0 \\ \alpha(e^x - 1) & \text{if } x < 0 \end{cases}$$

where λ and α are scaling parameters that ensure that the output of the activation function has a mean of 0 and a standard deviation of 1, thus enabling the activation function to prevent the vanishing/exploding gradient problem commonly encountered in deep neural networks. Specifically, λ is chosen to be the square root of the reciprocal of the expected value of the squared input of the activation function, and α is a constant that is related to the λ and the slope of the function.

Advantage:

- The SELU activation function is designed to ensure that the output of each layer of the network has a mean of 0 and a standard deviation of 1, which can help prevent the vanishing gradient problem and improve the training process.
- The self-normalizing property of the SELU activation function has been shown to lead to better performance on a wide range of neural network architectures and tasks, compared to other activation functions like ReLU and ELU.
- Because the SELU activation function ensures that the outputs of each layer are normalized, it eliminates the need for batch normalization, which can simplify the architecture of the network and improve its performance.

Disadvantage

- The computation of the SELU function involves exponential and logarithmic operations, which can make it computationally more expensive than other activation functions like ReLU and ELU.
- The self-normalizing property of the SELU activation function depends on the distribution of the weights and biases in the network. Therefore, the performance of the network may be sensitive to the choice of initialization method for the weights and biases.