**Step function:**

In mathematics, a step function (also called a Heaviside function) is a function that "steps up" or "steps down" at a certain point. The step function is usually denoted by the symbol u(t) or H(t), where t is the input variable.

The basic form of a step function is:

u(t) = { 0, t < 0

1/2, t = 0

1, t > 0 }

The step function takes on the value of 0 for all negative inputs, 1 for all positive inputs, and 1/2 at the point where the input is exactly zero.

Advantages:

* Simple and easy to compute.
* Useful for binary classification tasks.

Disadvantages:

* Not differentiable at zero, making it difficult to optimize using gradient-based methods.
* Gradient is either zero or undefined, which can also make it difficult to optimize.

**sigmoid function:**

The sigmoid function is a type of mathematical function that maps any input value to a value between 0 and 1. The most commonly used sigmoid function is the logistic function, which is defined as:

f(x) = 1 / (1 + e^(-x))

The sigmoid function has an S-shaped curve that starts at 0 and rises gradually to 1 as the input value increases. The function is symmetrical around x=0 and has a maximum slope of 0.25 at x=0.

advantages:

* Smooth gradient, making it good for optimization using gradient-based methods.
* Useful for binary classification tasks.

Disadvantages:

* Saturates when the input is large, causing gradients to vanish, which can make it difficult to train deep neural networks.
* Not zero-centered, which can lead to issues with gradient descent.

**Tanh function:**

The tanh function is a type of sigmoid function that maps any input value to a value between -1 and 1. Like the logistic function, the tanh function has an S-shaped curve that starts at -1 and rises gradually to 1 as the input value increases.The function is symmetrical around x=0 and has a maximum slope of 1 at x=0. The tanh function is often used as an activation function in neural networks to introduce non-linearity and to normalize input values.

The basic form of the hyperbolic tangent function (tanh) is:

tanh(x) = (e^x - e^(-x)) / (e^x + e^(-x))

where x is the input value and e is the mathematical constant known as Euler's number (approximately equal to 2.71828).

Advantages:

* Zero-centered, making it easier to train neural networks.
* Smooth gradient, making it good for optimization using gradient-based methods.

Disadvantages:

* Saturates when the input is large, causing gradients to vanish, which can make it difficult to train deep neural networks.

**Relu function:**

The basic form of the Rectified Linear Unit (ReLU) function is:

f(x) = max(0, x)

where x is the input value.The ReLU function returns the input value if it is positive, and 0 otherwise. This means that the ReLU function is a piecewise linear function that is zero for all negative values of x, and has a slope of 1 for all positive values of x.

The ReLU function is a popular activation function used in deep learning models. It is computationally efficient and has been shown to work well in practice, especially for image recognition tasks.

Advantages:

* Computationally efficient and easy to implement.
* Linear behavior for positive inputs, allowing for faster training and better convergence.
* Allows for sparse activation, which can help reduce overfitting.
* Less likely to suffer from the vanishing gradient problem compared to sigmoid and tanh functions.

Disadvantages:

* Outputs zero for negative inputs, which can cause the "dying ReLU" problem.
* Not differentiable at zero, causing issues with gradient-based optimization methods.
* Not zero-centered, which can lead to issues with gradient descent.

**Elu function:**

The basic form of the Exponential Linear Unit (ELU) function is:

f(x) = x if x >= 0, alpha \* (e^x - 1) if x < 0

where x is the input value, and alpha is a hyperparameter that controls the function's behavior for negative inputs.The ELU function is similar to the ReLU function, but it has a non-zero output for negative inputs, which can help prevent the "dying ReLU" problem. The function is also differentiable for all input values, which makes it well-suited for use in gradient-based optimization algorithms.The ELU function has a negative saturation regime where the output approaches a limit as x goes to negative infinity, which can help improve the robustness of deep learning models.

Advantages:

* Continuously differentiable, making it suitable for optimization using gradient-based methods.
* Has a negative region, allowing for activation in both positive and negative ranges.
* Solves the "dying ReLU" problem by preventing neurons from becoming permanently inactive.

Disadvantages:

* Computationally more expensive compared to ReLU.
* May lead to overfitting if not properly tuned.

**Selu function:**

The basic form of the Scaled Exponential Linear Unit (SELU) function is:

f(x) = lambda \* (x if x >= 0 else (alpha \* (e^x - 1)))

where x is the input value, lambda and alpha are scaling parameters, and e is Euler's number (approximately equal to 2.71828).The SELU function is similar to the ELU function, but it has a scaling factor that helps preserve the mean and variance of the input values during training. This can help improve the performance of deep learning models and make them more robust to different types of input data.

The SELU function has several important properties, including:

Self-normalization: When certain conditions are met (such as the input data having zero mean and unit variance), the outputs of the SELU function tend to have zero mean and unit variance as well, which can help improve the stability of deep learning models during training.

Continuously differentiable: The SELU function is differentiable for all input values, which makes it well-suited for use in gradient-based optimization algorithms.

Zero-centered: The SELU function is centered around zero, which can help improve the performance of deep learning models by reducing the bias introduced by activation functions that are not centered around zero.

The SELU function is a popular activation function used in deep learning models, particularly for image and speech recognition tasks.

Advantages:

* Designed to be self-normalizing, which can improve performance and convergence in deep neural networks.
* Allows for activation in both positive and negative ranges.

Disadvantages:

* Computationally more expensive compared to ReLU.
* Requires specific weight initialization and activation function scaling to work properly.