Name:ISLAM,MD. JUNIADUL
Id:20-41950-1
Sec:A

Report on Activation Function

An activation function is a mathematical function that is applied to a neuron's output in artificial neural networks and machine learning in order to bring nonlinearity into the network's output. Based on information from other neurons in the network, the activation function aids in deciding whether or not a neuron should fire. I'll go through six different activation functions in this study, their mathematical foundations, benefits, and drawbacks.

1.Step function:

The step function, also known as the Heaviside step function, is a simple activation function commonly used in neural networks. It is defined mathematically as follows:

$$\begin{array}{rll} 0, & if & x < 0 \\ H(x) = \&1, & if & x \geq 0 \end{array}$$

The step function has a few advantages and disadvantages:

Advantages:

- The step function is simple to implement and can be computed quickly.
- It is a binary function, which makes it useful for binary classification problems.
- It is easy to interpret and understand, which makes it useful for educational purposes.

Disadvantages:

- The step function is not differentiable at x=0, which can cause problems when training neural networks using gradient descent algorithms.
- It is not suitable for regression problems or any task that requires the output to be a continuous value.
- The step function can suffer from the problem of vanishing gradients, where the gradients of the function become too small to be useful for training the network.

Thus, despite being a helpful activation function in some situations, the step function is not frequently used in conventional neural networks due to its drawbacks. Due to their smoother gradients and capacity to handle a larger range of issues, other activation functions, such as the sigmoid or ReLU functions, are often favored.

2. Sigmoid function:

The sigmoid function is a mathematical function that maps any input 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))

where x is the input to the function.

Advantages:

- Sigmoid functions are easy to work with and can be easily differentiated, which makes them useful in optimization algorithms like gradient descent.
- They are widely used in neural networks to map the output of a neuron to a probability distribution.
- Sigmoid functions are bounded, meaning that their outputs are always between 0 and 1, which can be useful in certain applications like probability calculations.

Disadvantages:

- Sigmoid functions are prone to saturation, which means that for large values of x, the output of the function becomes very close to 1, making it difficult for the function to learn further.
- The gradient of the sigmoid function becomes very small for large values of x, which can lead to slow convergence in optimization algorithms.
- Sigmoid functions are not symmetric around zero, which means that they can introduce bias into the output of a model.

3. Tanh function:

A mathematical function called the hyperbolic tangent function (tanh) converts input values into output values between -1 and 1. The equation is as follows:

tanh(x) = (e^x - e^(-x)) / (e^x + e^(-x)) where e is the mathematical constant approximately equal to 2.71828, and x is the input value.

Advantages:

- Like the sigmoid function, tanh is also a smooth function that can be easily differentiated, which makes it useful in training neural networks using backpropagation.

- It is a zero-centered function, which means that its outputs are centered around zero. This can help in preventing vanishing gradients during the training of deep neural networks.
- Tanh is bounded between -1 and 1, which makes it useful for normalization of data that has a wide range of values.

Disadvantages:

- Like the sigmoid function, tanh can suffer from the vanishing gradient problem, which can make it difficult to train deep neural networks.
- Tanh is not monotonic, which means that its derivative is not always positive or negative. This can make it more difficult to optimize using some optimization techniques.
- The output of tanh is not sparse, which means that it can be less efficient than other activation functions in terms of memory and computation requirements.

Tanh function, which offers advantages over other functions like the sigmoid function, is a valuable activation function in neural networks overall. It does, however, have significant restrictions that must be taken into account while applying it in various situations.

4. Relu function:

The ReLU (Rectified Linear Unit) function is a commonly used activation function in neural networks. It is defined as:

$f(x) = \max(0, x)$ where x is the

input to the function.

Advantages:

- The ReLU function is computationally efficient compared to other activation functions.

- It has been shown to work well in many types of neural networks, including deep neural networks.
- It is easy to implement and interpret.
- The ReLU function has a sparse output, which can help prevent overfitting.

Disadvantages:

- The ReLU function can suffer from the "dying ReLU" problem, where some neurons stop producing any output due to the input being negative. This can cause the network to become less expressive and impact its performance.
- The ReLU function is not symmetric, which can make it difficult to use in certain types of neural networks.
- The ReLU function is not differentiable at x=0, which can cause issues in some optimization algorithms that rely on gradient information.

In conclusion, the ReLU function is a well-liked activation function since it is computationally effective and useful in many different kinds of neural networks. It does, however, have certain drawbacks, such as the potential for the "dying ReLU" problem and difficulties with optimization techniques.

5. ELU function:

The ELU (Exponential Linear Unit) step function is an activation function used in artificial neural networks. It is a variant of the ReLU (Rectified Linear Unit) function and is defined as:

ELU(x) = { x if x >= 0, alpha * (exp(x) - 1) if x < 0 }

where alpha is a hyperparameter that determines the negative saturation value.

Advantages:

- Smoother gradients: The ELU function has a continuous derivative, which helps avoid the vanishing gradient problem that can occur with other activation functions like the sigmoid function.
- Better performance: Empirical evidence suggests that ELU activation functions can lead to better performance in deep neural networks compared to other activation functions.
- Faster convergence: The ELU function has been shown to help networks converge faster than other activation functions.

Disadvantages:

- Computationally expensive: The ELU function requires computing the exponential function, which can be computationally expensive.
- Unstable for large negative inputs: The ELU function can be unstable for large negative inputs, which can lead to numerical instability in the network.
- Not as well-known: The ELU function is not as well-known as other activation functions like the sigmoid and ReLU functions, which can make it harder to find resources and support for implementing it in neural networks.

6. SELu function:

The SELu (Scaled Exponential Linear Unit) activation function is a type of activation function used in neural networks. Unlike other activation functions, such as ReLU, which have a hard cutoff at zero, SELu has a smooth, continuous curve.

The formula for the SELu activation function is:

f(x) = 1.0507 * (e^x - 1), x < 0
f(x) = x, x >= 0

Advantages:

- The SELu activation function is a self-normalizing function, which means that the output of each neuron has zero mean and unit variance, regardless of the input distribution. This can help to reduce the problem of vanishing/exploding gradients, which can occur in deep neural networks.
- SELu has been shown to perform better than other activation functions, such as ReLU and tanh, on a variety of deep learning tasks, including image classification and speech recognition.

Disadvantages:

- The SELu activation function is computationally more expensive than other activation functions, such as ReLU, because it involves exponentiation.

- SELu may not work well on all types of data. While it has been shown to work well on many deep learning tasks, it may not be the best choice for all types of data, and some experimentation may be necessary to determine the best activation function for a particular task.