Name: Md. Abrar Hamim

ID:20-42832-1

Activation Function

An activation function is a mathematical function used in neural networks to introduce non-linearity into the output of a neuron. It takes the weighted sum of inputs to the neuron and applies a transformation to produce the neuron's output. I will discuss six different activation functions, their mathematical formulas, advantages, and disadvantages.

Step function

The step function is a mathematical operation that yields a binary output of 0 or 1 from a real-valued input. The mathematical formula for the step function is as follows:

f(x) = 0 if x < 0

f(x) = 1 if x >= 0

Advantage:

1. The step function is a simple function that is easy to understand and implement. It is often used in introductory courses on neural networks as an example of an activation function.
2. The step function produces a binary output of 0 or 1, which can be useful in certain applications, such as binary classification problems.
3. The step function produces a binary output of 0 or 1, which can be useful in certain applications, such as binary classification problems.
4. The step function produces a binary output of 0 or 1, which can be useful in certain applications, such as binary classification problems.

Disadvantage:

1. The step function is a discontinuous function, which means that it has a sharp transition from 0 to 1 at the threshold value of zero. This discontinuity makes it difficult to use in many neural network architectures and optimization algorithms that require continuous functions.
2. The step function is non-differentiable at the threshold value of zero, which means that it has no defined derivative at this point. This non-differentiability makes it difficult to use in many gradient-based optimization algorithms that require the calculation of derivatives.
3. The step function has limited expressive power since it can only produce binary outputs of 0 or 1. This limitation makes it unsuitable for many applications that require more fine-grained outputs.
4. The step function has a constant gradient of zero for all inputs less than the threshold value of zero and a constant gradient of one for all inputs greater than or equal to the threshold value.

SIGMOID FUNCTION

A mathematical function called the sigmoid function generates a sigmoid-shaped output between 0 and 1 from a real-valued input. The mathematical formula for the step function is as follows:

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

Advantage:

1. The sigmoid function maps any real-valued input to a value between 0 and 1, which can be interpreted as a probability. This makes it particularly useful for binary classification problems, where the output of the sigmoid function can be interpreted as the probability that the input belongs to one of two classes.
2. The sigmoid function has a bounded output between 0 and 1, which can be useful in certain applications where the output needs to be constrained to a specific range.
3. The sigmoid function introduces nonlinearity into the output of a neural network, which is important for enabling the network to model complex relationships between inputs and outputs.
4. The sigmoid function can be particularly useful in shallow neural networks, where it can help produce stable and accurate results.
5. The sigmoid function produces a smooth and continuous output, which makes it easy to compute gradients and use gradient-based optimization algorithms like backpropagation.

Disadvantage:

1. The sigmoid function has a tendency to saturate for very large or very small input values, which can cause the gradient of the loss function with respect to the weights to become very small or even zero.
2. The output of the sigmoid function is not zero-centered, which can make it more difficult to train neural networks using certain optimization algorithms, such as stochastic gradient descent with momentum.
3. The sigmoid function can be computationally expensive to compute, especially when compared to other activation functions like the rectified linear unit (ReLU) or its variants.
4. The sigmoid function is not sparse, which means that all neurons in the layer will produce an output for any input value. This can lead to overfitting in some cases, especially when the network is too large or the data is too noisy.

Tanh function:

The tanh function, also known as the hyperbolic tangent function, is a mathematical function that accepts a real-valued input and yields an output that is in the range of -1 and 1. The mathematical formula for the step function is as follows:

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

Advantage:

1. The tanh function produces an output that is centered around zero, which can be useful in applications where the input data is also zero-centered. This property can help improve the convergence of neural network models, particularly in cases where the data is symmetric around zero.
2. The tanh function produces a smooth and continuous output, which makes it easy to compute gradients and use gradient-based optimization algorithms like backpropagation. The tanh function has a well-defined derivative that can be expressed in terms of the function itself, which simplifies the computation of gradients.
3. The tanh function has a bounded output between -1 and 1, which can be useful in certain applications where the output needs to be constrained to a specific range.
4. The tanh function introduces nonlinearity into the output of a neural network, which is important for enabling the network to model complex relationships between inputs and outputs.

Disadvantage:

1. The tanh function has a tendency to saturate for very large or very small input values, which can lead to vanishing gradients and slow down learning in deep neural networks.
2. The output of the tanh function is bounded between -1 and 1, which may not be suitable for certain types of data that have a different range.
3. The tanh function is not sparse, which means that all neurons in the layer will produce an output for any input value.
4. While the tanh function can alleviate the vanishing gradients problem to some extent in deep neural networks, it is still susceptible to this problem for very deep networks with many layers.

Relu function:

ReLU function returns 0 for any negative input value, and the input value itself for any non-negative input value.

$f(x) = \max(0, x)$

Advantages:

1. The ReLU function is very simple to compute and does not require any complex mathematical operations like the exponential function required by the sigmoid and tanh functions.
2. The ReLU function can produce a sparse output, which means that only a subset of neurons in a layer will produce an output for any given input value. This can help reduce overfitting and improve generalization performance in some cases.
3. The ReLU function is very efficient to compute and can be implemented in hardware with very low power consumption.

Disadvantages:

1. The ReLU function can suffer from the problem of "dead neurons", where the neuron becomes permanently inactive and produces an output of 0 for any input value.
2. The ReLU function is unbounded above, which means that it can produce very large output values for large input values. This can lead to numerical instability and make it harder to train the network.

3. The ReLU function is not suitable for data that has negative input values, as it simply sets them to 0.
4. The ReLU function can be sensitive to the initialization of the weights in the network, as it can cause a large fraction of the neurons to become inactive and produce a 0 output.

Elu function:

The Exponential Linear Unit (ELU) function is an activation function that is similar to the ReLU function but has some advantages over it. It is defined as:

f(x) = x, x > 0

alpha * (exp(x) - 1), x <= 0

Advantages:

1. 1.The ELU function is smooth and differentiable everywhere, which can help improve the training of deep neural networks.
2. ELU function has a non-zero mean output, which can help reduce the bias shift problem that can occur with the ReLU function.
3. The ELU function saturates to a negative output for very negative input values, which can help the network to resist the "exploding gradients" problem that can occur in deep neural networks.

Disadvantages:

1. The ELU function is more computationally expensive than the ReLU function, as it involves an exponential operation.
2. Although the ELU function has some theoretical advantages over the ReLU function, there is limited empirical evidence that it consistently outperforms the ReLU function on a wide range of tasks.

SELU FUNCTION:

The Scaled Exponential Linear Unit (SELU) function is an activation function for artificial neural networks. It is a self-normalizing activation function that has shown promise in improving the stability and convergence of deep neural networks. The SELU function is defined as:

f(x) = lambda * (exp(x) - 1), x < 0

x, x >= 0

Advantages:

1. The SELU function has a property called "self-normalization" that can help improve the stability and convergence of deep neural networks.
2. The SELU function has a mean output of 0 and a standard deviation of 1, which can help reduce the internal covariate shift problem that can occur in deep neural networks.
3. The SELU function has been shown to improve the performance of deep neural networks on a wide range of tasks, especially when compared to other activation functions like the ReLU and ELU functions.

Disadvantages:

1. The SELU function has strict requirements for the initialization of the weights in the network, which can make it more difficult to use than other activation functions.
2. Although the SELU function has shown promising results in some studies, there is still limited empirical evidence on its effectiveness and generalizability across different architectures and tasks.
3. The self-normalizing property of the SELU function makes it less compatible with certain types of networks, such as convolutional neural networks, which typically have different weight initialization requirements.