American International University-Bangladesh

Report on Activation Functions

Computer Vision and Pattern Recognition

Section: B

Submitted By,

Zerin Hasan Sahosh

ID: 20-43744-2

Department of CSE

AIUB

Submitted To,

Dr. Debajyoti Karmaker

Assistant Professor

Department of CSE

AIUB

Activation Function

An activation function is a mathematical function that is applied to the output of a neural network layer to introduce non-linearity into the model. The purpose of the activation function is to transform the input signal into an output signal that can be interpreted by the next layer of the neural network.

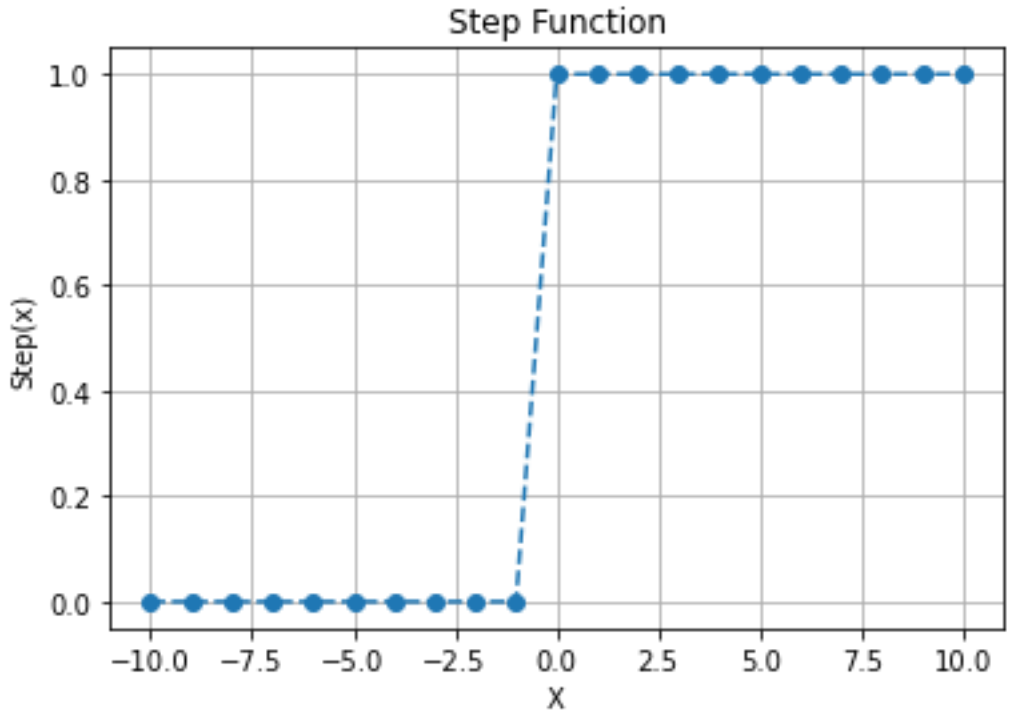
The activation function determines whether the neuron should be activated or not based on whether the weighted sum of inputs received by the neuron is above a certain threshold. If the input exceeds the threshold, the neuron is "activated" and sends its output signal to the next layer of neurons.

There are several types of activation functions, including the popular ones like Step, Sigmoid, Elu ReLU, Tanh, and Selu. Each has its own advantages and disadvantages and is suitable for different types of neural network architectures and tasks.

1. Step Function

A step function is a type of activation function used in artificial neural networks, which takes an input and returns an output of either 0 or 1 based on a threshold. It is also known as the Heaviside step function or the unit step function.

The step function takes an input value and compares it to a threshold value. If the input value is greater than or equal to the threshold, the function returns 1. Otherwise, it returns 0. The step function is a binary function, meaning it has only two possible output values.

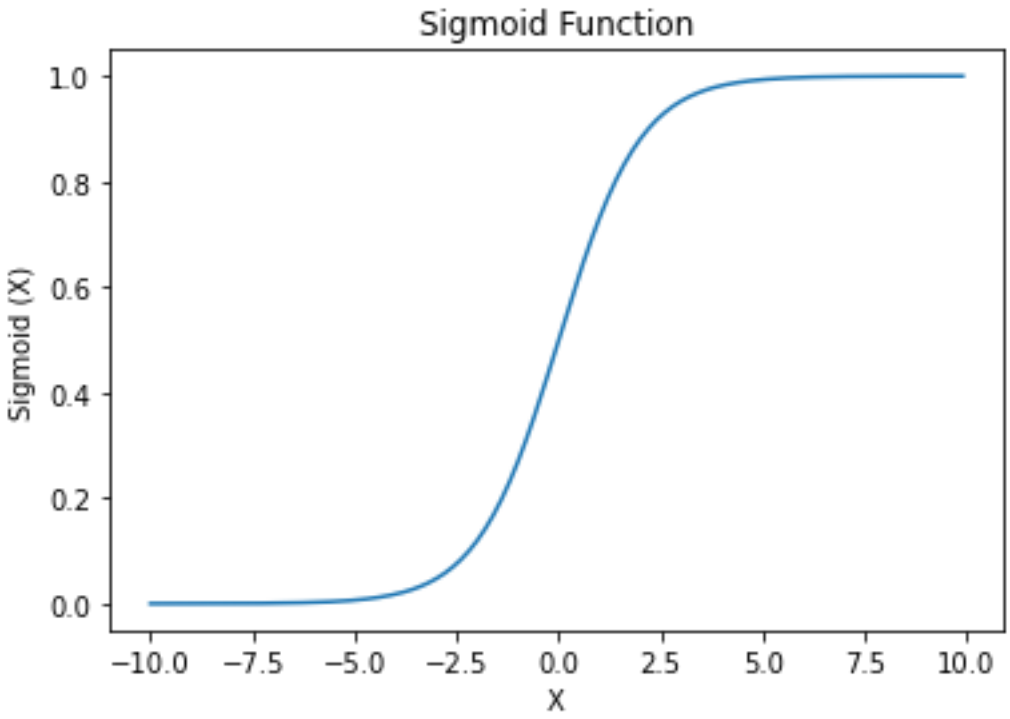


2. Sigmoid Function

The sigmoid function is a mathematical function commonly used in machine learning and neural networks. It is a type of activation function that maps any input value to a value between 0 and 1. The sigmoid function has an S-shaped curve and is defined mathematically as: f(x) = 1 / (1 + e^(-x))

The sigmoid function is useful in machine learning because it can be used to model the probability of an output being in a certain class, where the output value ranges from 0 to 1. It is also used in neural networks as an activation function to introduce nonlinearity in the network and allow the network to learn complex relationships between inputs and outputs.

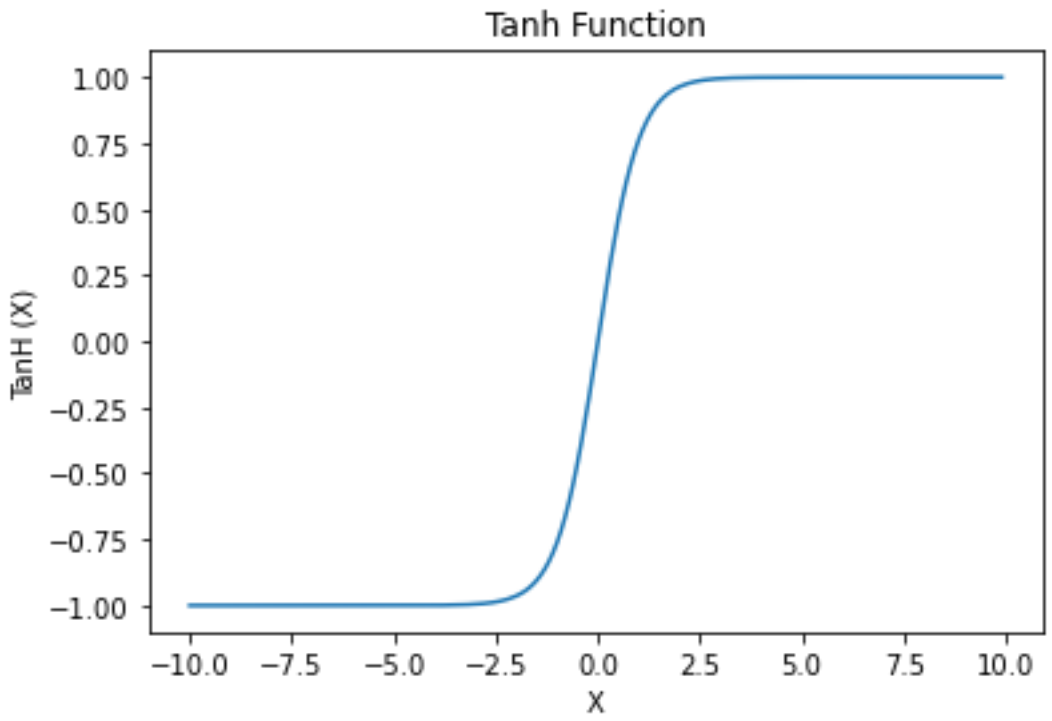
One of the properties of the sigmoid function is that its output values are always between 0 and 1, which can be interpreted as probabilities.



3. Tanh Function

The tanh function, short for hyperbolic tangent function, is a mathematical function commonly used in machine learning and neural networks. It is another type of activation function that maps any input value to a value between -1 and 1. The tanh function has a similar S-shaped curve as the sigmoid function, but its output ranges from -1 to 1, and is defined mathematically as: tanh(x) = (e^x - e^(-x)) / (e^x + e^(-x))

One of the properties of the tanh function is that it is symmetric around the origin, meaning that tanh(-x) = -tanh(x). Another property is that the output values saturate at the extremes, meaning that when x is very large or very small, the output value approaches -1 or 1 respectively.

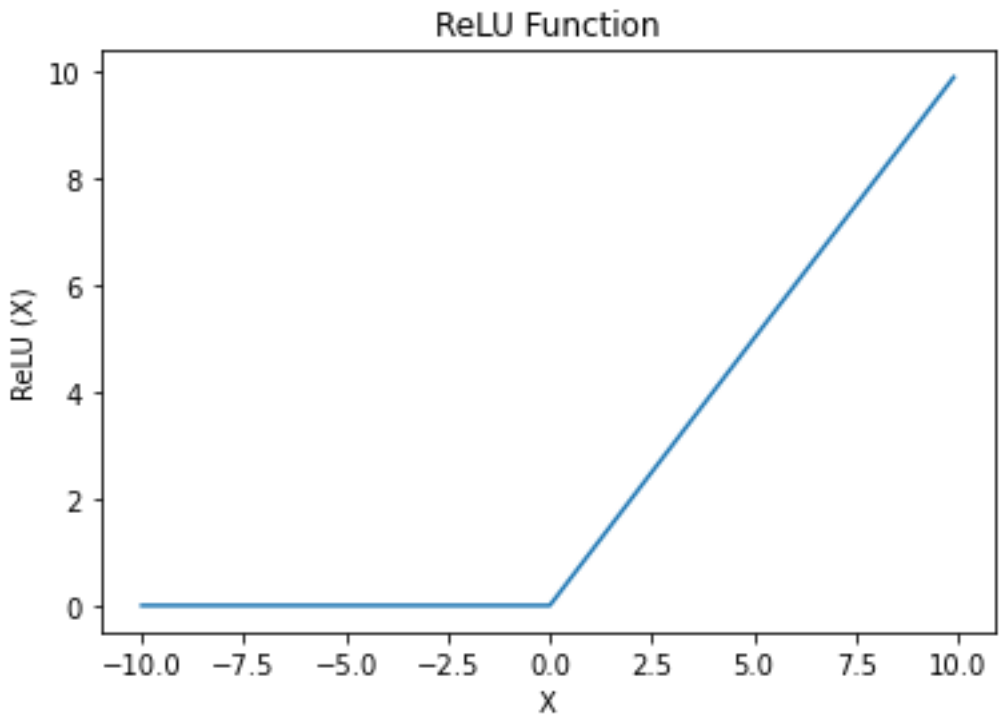


4. Relu Function

The ReLU (Rectified Linear Unit) function is a popular activation function used in neural networks and deep learning. The ReLU function is a simple piecewise linear function that maps any input value less than or equal to zero to zero, and any input value greater than zero to the same value. Mathematically, the ReLU function can be defined as: f(x) = max(0, x)

The ReLU function is often preferred over other activation functions, such as the sigmoid and tanh functions, because it is computationally efficient to compute and can speed up training of deep neural networks. It is also less prone to the vanishing gradient problem, which can occur when using other activation functions that saturate at the extremes.

However, one of the limitations of the ReLU function is that it can cause "dead" neurons, where the neuron outputs 0 for all inputs, effectively "turning off" the neuron.



5. Elu Function

The ELU (Exponential Linear Unit) function is an activation function used in neural networks and deep learning. Like the ReLU function, it is computationally efficient and addresses the vanishing gradient problem, but it has a few advantages over the ReLU function.

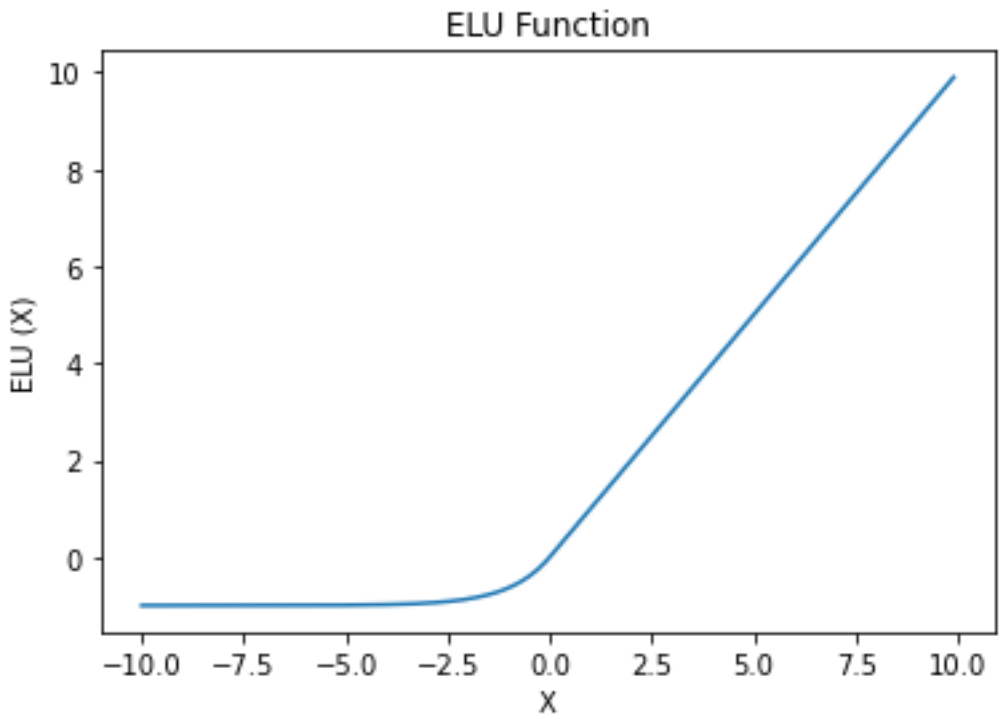
The ELU function is defined as:

f(x) = x, x >= 0

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

The ELU function is similar to the ReLU function, but it has a nonzero output for negative inputs, which can help prevent "dead" neurons. Additionally, the ELU function is smooth and continuously differentiable, which can make it easier to optimize in some cases.

One of the drawbacks of the ELU function is that it is more computationally expensive to compute than the ReLU function, since it involves exponentiation. However, the main disadvantage of the ELU function is that it is more computationally expensive than the ReLU function, since it involves computing an exponential function for negative inputs. As a result, the ReLU function is still the most commonly used activation function in deep learning.



6. Selu Function

SELU (Scaled Exponential Linear Unit) is an activation function for deep neural networks that was introduced in 2017. It is a self-normalizing activation function, which means that it tends to preserve the mean and variance of the activations across layers, even as the inputs propagate through the network.

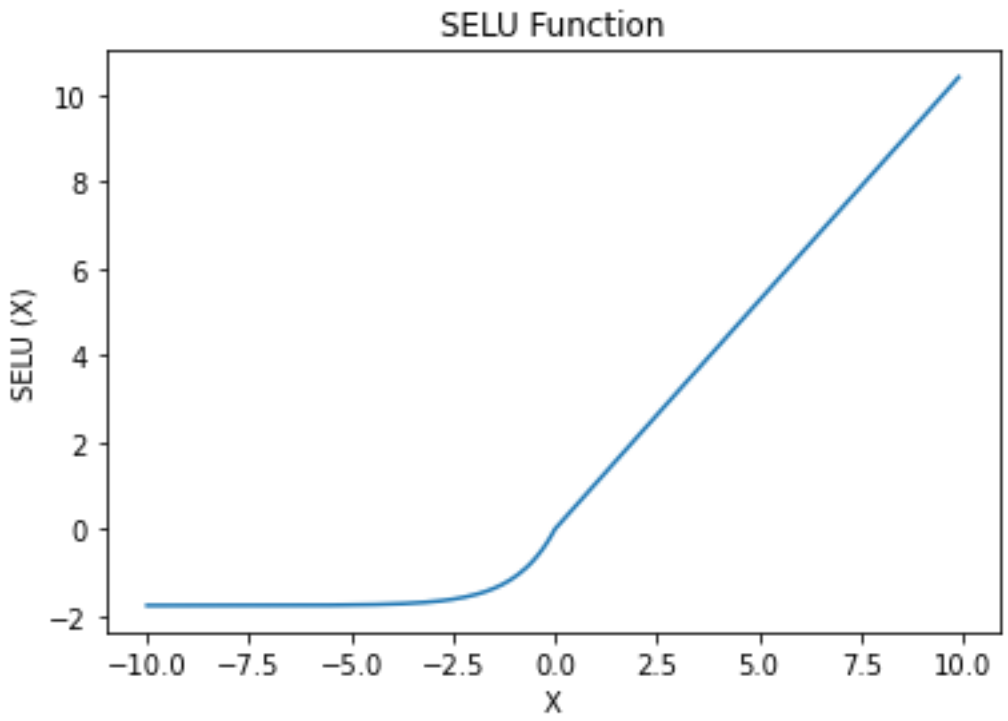
The SELU function is defined as:

f(x) = lambda * (max(0, x) + alpha * (exp(min(0, x)) - 1))

The main advantage of the SELU function is its ability to improve the performance and stability of deep neural networks, particularly in cases where the data has a complex structure or high dimensionality. The self-normalizing property of the SELU function helps to mitigate the vanishing and exploding gradient problems that can arise in deep networks and lead to poor performance or unstable training.

However, the SELU function also has some limitations. It requires careful initialization of the network weights, and it may not perform as well on certain types of data or in certain architectures. Additionally, it is more computationally expensive than some other activation functions, such as ReLU.

Overall, the SELU function is a promising option for deep neural networks that require high performance and stability, but it may not be suitable for all applications.


SELU Function
SELU (X)
X
10
8
6
4
2
0
−2
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0