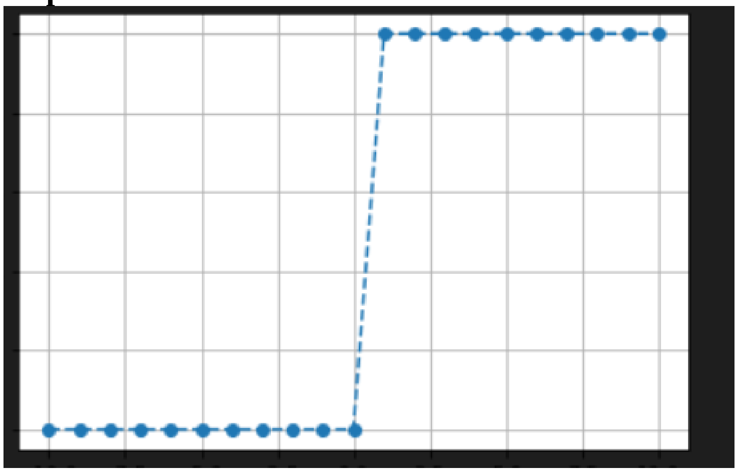
Activation functions

1. **Step Function**:



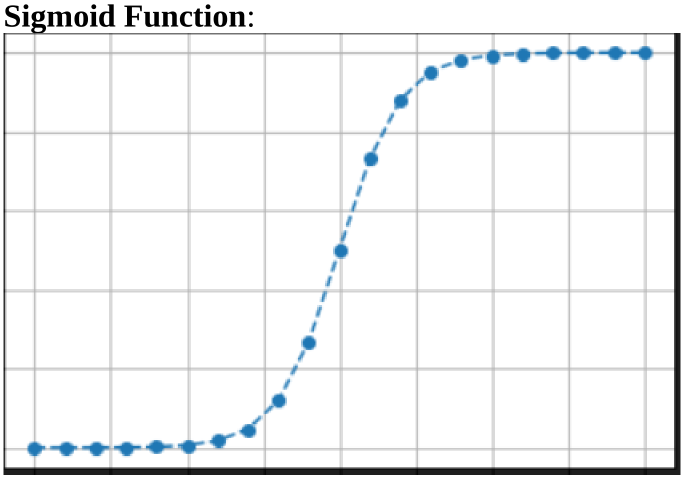
Advantages:

- Easy to compute and implement.
- Simple and easy to understand.
- Useful in modeling real-world phenomena where the value of the function changes abruptly at specific points.

Disadvantages:

- Non-differentiable function
- Limited accuracy and applicability.
- It cannot be used in backpropagation algorithms for deep learning
- Discontinuous nature may lead to convergence problems.
- Its output is discrete, which makes it difficult to map to probabilities or perform other continuous operations.

2. **Sigmoid Function**:



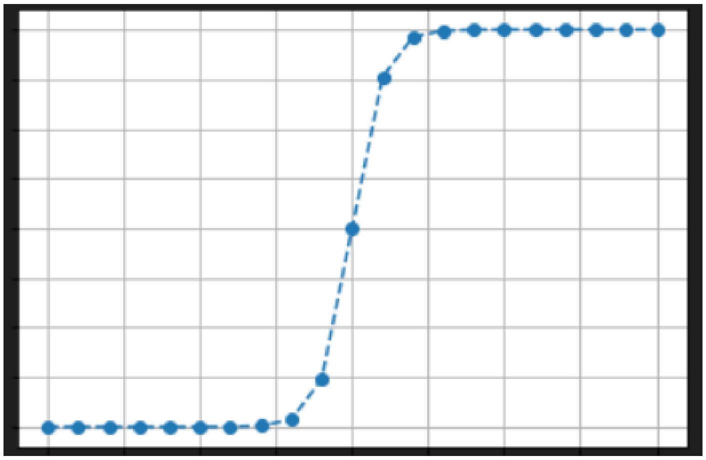
Advantages:

- Smooth and continuous.
- Outputs are bounded between 0 and 1, making it useful for binary classification problems.
- Easy to compute and implement.

Disadvantages:

- Vanishing gradient problem may occur.
- Outputs outside the range [-2, 2] are close to zero, leading to a saturation of the gradients and slower learning.
- Its output is not zero-centered. This means that the majority of the values in the output will be either close to 0 or close to 1, with few values in between. This can make it difficult to interpret, and can lead to issues such as saturation and diminishing gradients in deep learning algorithms.

3. **Tanh Function:**



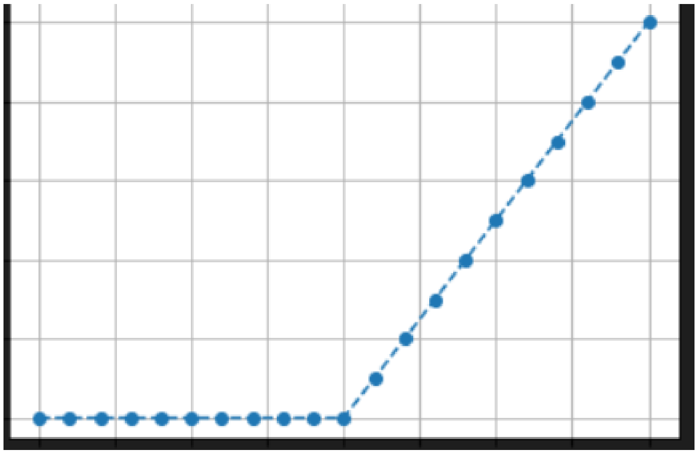
Advantages:

- Tanh function is that its output is centered around zero
- Smooth and continuous.
- Outputs are bounded between -1 and 1, making it useful for symmetric data.
- Easy to compute and implement.

Disadvantages:

- Vanishing gradient problem may occur.
- Similar saturation problem as the sigmoid function.
- It can be computationally expensive, which can limit its use in machine learning algorithms

4. **Rectified Linear Units (ReLU) Function: Advantages:**

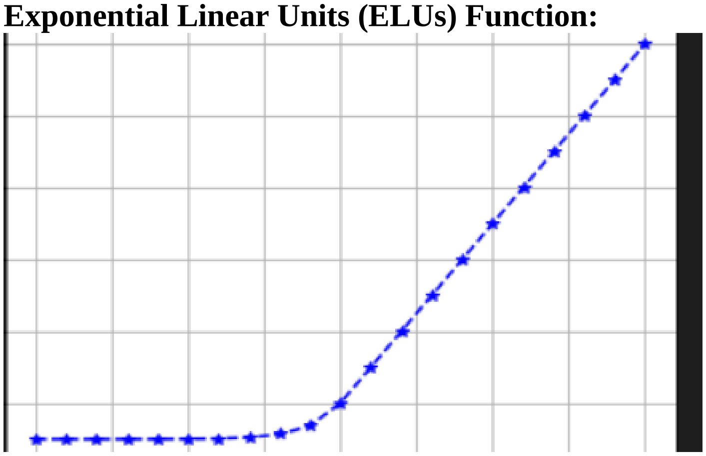


- Simple and computationally efficient.
- It is a smooth and continuous function, which makes it suitable for use in gradient-based optimization algorithms
- Fast convergence due to the non-saturating nature of the function.
- Effective in preventing vanishing gradients.

Disadvantages:

- Its output is not always bounded between -1 and 1, which can make it difficult to map to probabilities
- Outputs are not bounded, leading to the exploding gradients problem.
- Outputs can be fragile to noise.

5. **Exponential Linear Units (ELUs) Function:**



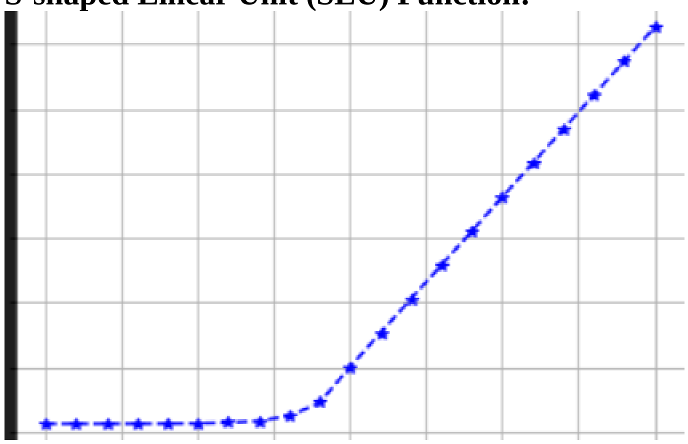
Advantages:

- Smooth and continuous.
- Outputs are negative for negative inputs, making it more biologically plausible.
- Effective in preventing vanishing gradients.
- It is monotonic, which means that its output either increases or decreases as the input increases, making it easier to interpret.

Disadvantages:

- Its output is not always bounded between -1 and 1, which can make it difficult to map to probabilities.
- Outputs are not bounded, leading to the exploding gradients problem.
- More computationally expensive than ReLU.

6. **S-shaped Linear Unit (SLU) Function:**



Advantages:

- Smooth and continuous.
- It is suitable for use in gradient-based optimization algorithms.
- Outputs are bounded between 0 and 1, making it useful for binary classification problems.
- Effective in preventing vanishing gradients.

Disadvantages:

- Its output is not always bounded between -1 and 1, which can make it difficult to map to probabilities
- Not as widely used and tested as other activation functions.
- May require more computation and time compared to other functions.

The ELU function is similar to the ReLU function for **x > 0**, but it has a smooth curve for **x < 0**, which allows it to produce negative output values. This can be useful in certain types of neural networks, such as those used for image processing or natural language processing. The SELU activation function is similar to the ELU function, but with a normalization step to ensure that the mean and variance of the output are approximately constant during training. One advantage of the SELU function over the ELU function is that it has been shown to produce better results on certain types of neural networks, especially those with many layers. This is due to the normalization property, which helps prevent the vanishing/exploding gradient problem that can occur during training. the SELU function is similar to the ELU function for **x > 0**, but it has a steeper curve for **x < 0**, which allows it to produce negative output values more easily. Additionally, the function has a normalization property, which helps prevent the vanishing/exploding gradient problem during training. This can be especially useful for deep neural networks with many layers.