Activation functions

1. **Step Function**:

The step function accepts an input and, depending on whether the input is larger than or equal to a threshold value, returns a binary output of 0 or 1. The step function's graph is a straight line that changes from 0 to 1 at the threshold value. The output is 0 if the threshold value is less than it, and 1 if the threshold value is greater. The step function is discontinuous, which makes it difficult to work with in some situations because it jumps sharply at the threshold value.

Advantages:

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

Disadvantages:

- Limited accuracy and applicability.
- Discontinuous nature may lead to convergence problems.

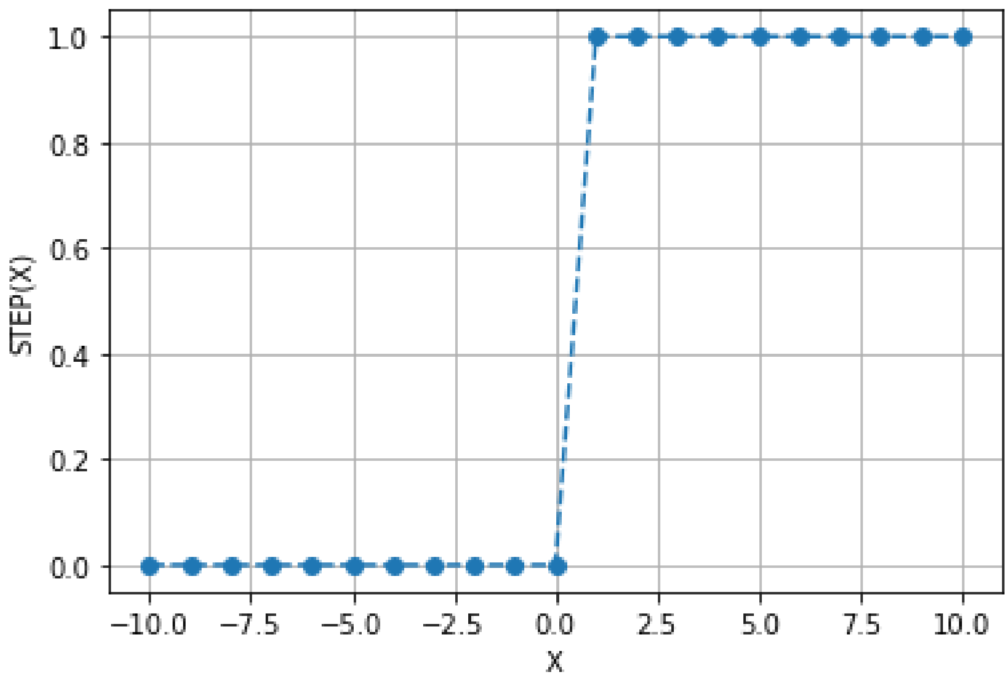

Figure 1: Step function Graph

2. **Sigmoid Function**:

The sigmoid activation function is helpful for tasks where the output of a model needs to be understood as a probability. It maps any input to a value between 0 and 1. As can be seen from the graph, the sigmoid function is a continuous, smooth curve that rises steadily over time from 0 to 1. Its gradual slope near the middle makes it an excellent option for representing probability. Deep neural networks that use this function may find it difficult to train due to the "vanishing gradient" problem, which occurs as the gradient of the function approaches 0 as the input goes away from the inflection point.

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.

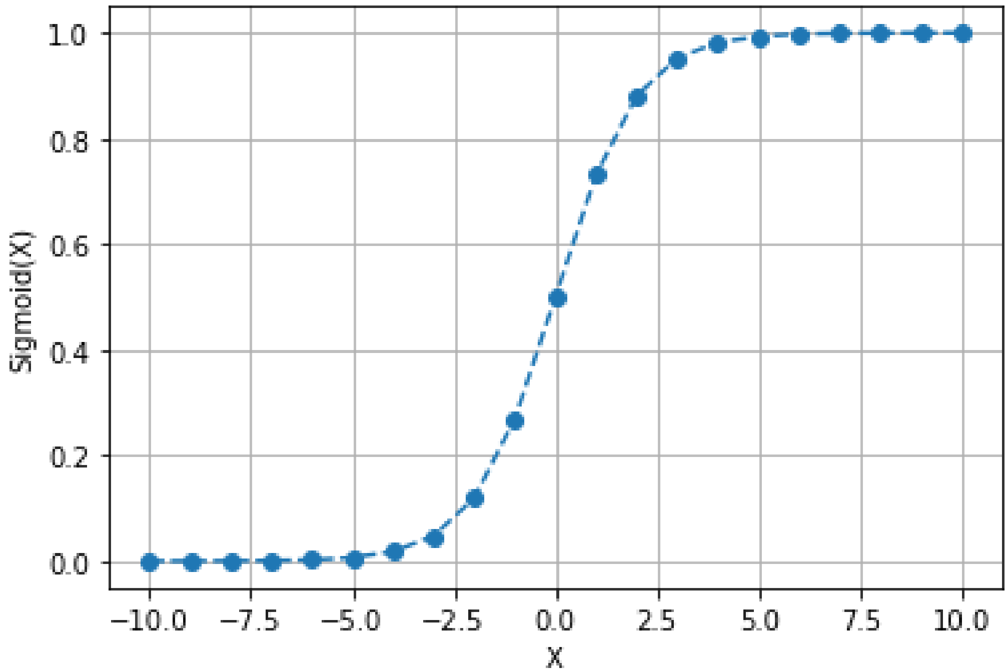

Figure 2: Sigmoid function Graph

3. **TanH Function**:

The Tan(h), Tan(H), and tanh(h) functions all relate to the same thing: the hyperbolic tangent activation function. Its shape, an S, is highly reminiscent of the sigmoid activation function. Any real number can be fed into the function, and it will return a number between -1 and 1. The output value will be closer to 1 for bigger or positive inputs and closer to -1 for smaller or negative inputs.

Advantages:

- 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.

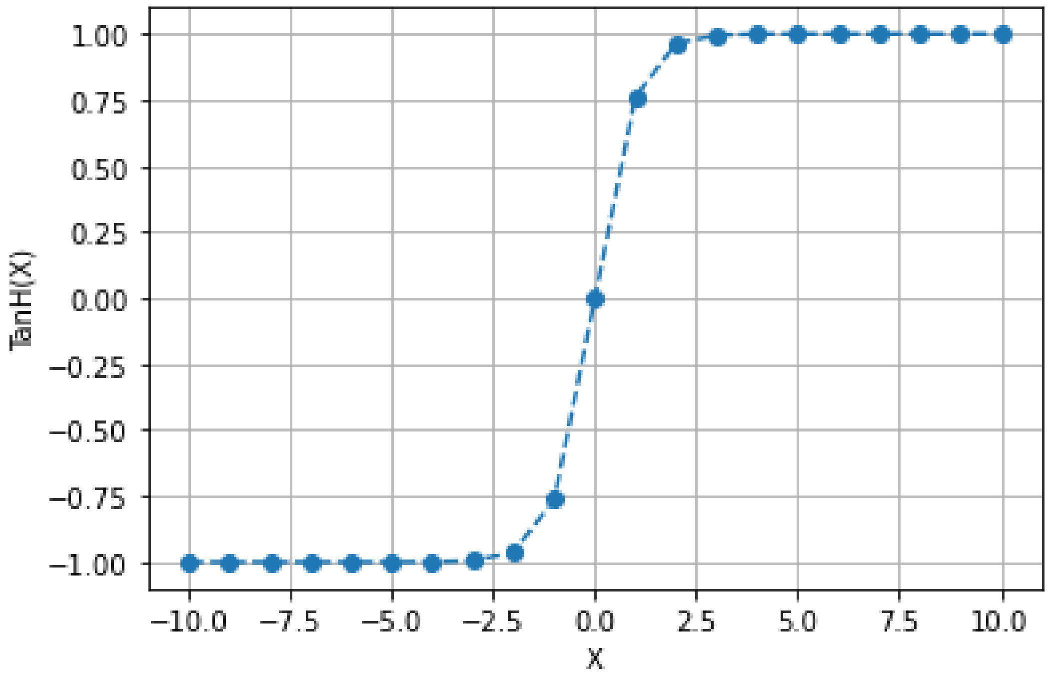

Figure 3: TanH function Graph

4. **Rectified Linear Units (ReLU) Function:**
   A Rectified Linear Unit, or ReLU, is a type of linear measurement. It appears and behaves like a linear algorithm, but it actually gives the same benefits as Sigmoid while being more efficient. To put it another way, ReLU can go anywhere from 0 to infinity.

Advantages:
- Simple and computationally efficient.
- Fast convergence due to the non-saturating nature of the function.
- Effective in preventing vanishing gradients.

Disadvantages:
- Outputs are not bounded, leading to the exploding gradients problem.
- Outputs can be fragile to noise.

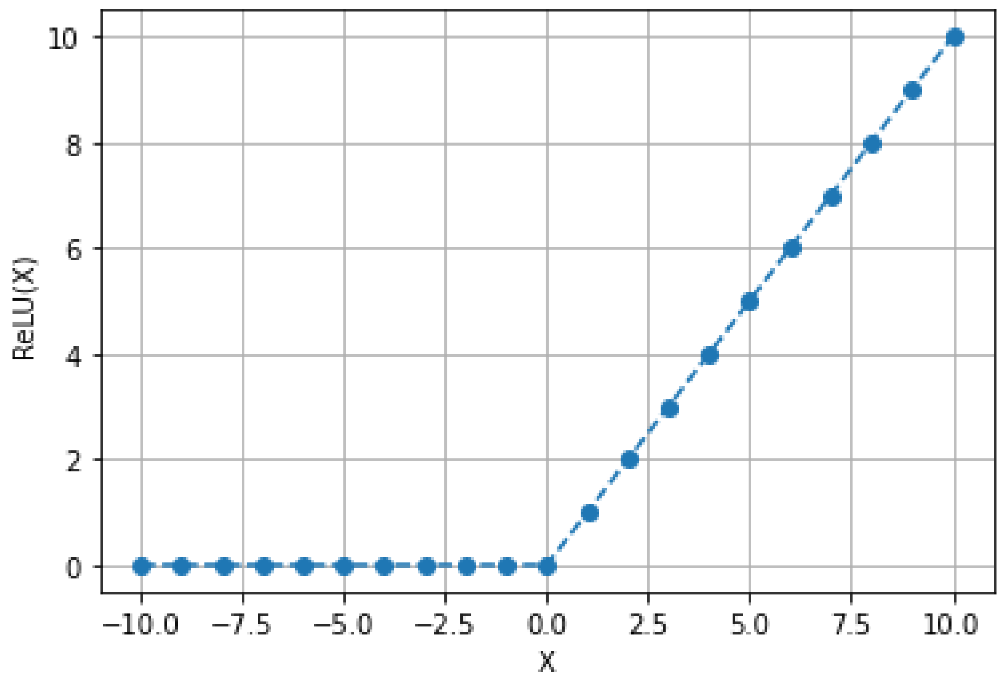

Figure 4: ReLU function Graph

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.

Disadvantages:

- Outputs are not bounded, leading to the exploding gradients problem.
- More computationally expensive than ReLU.

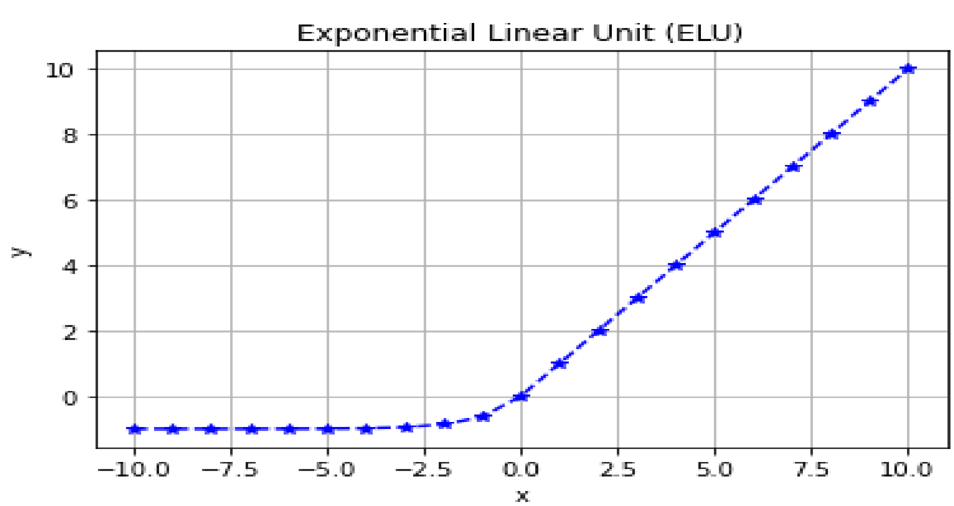

Figure 5: ELU function Graph

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

Advantages:

- Smooth and continuous.
- Outputs are bounded between 0 and 1, making it useful for binary classification problems.
- Effective in preventing vanishing gradients.

Disadvantages:

- Not as widely used and tested as other activation functions.
- May require more computation and time compared to other functions.

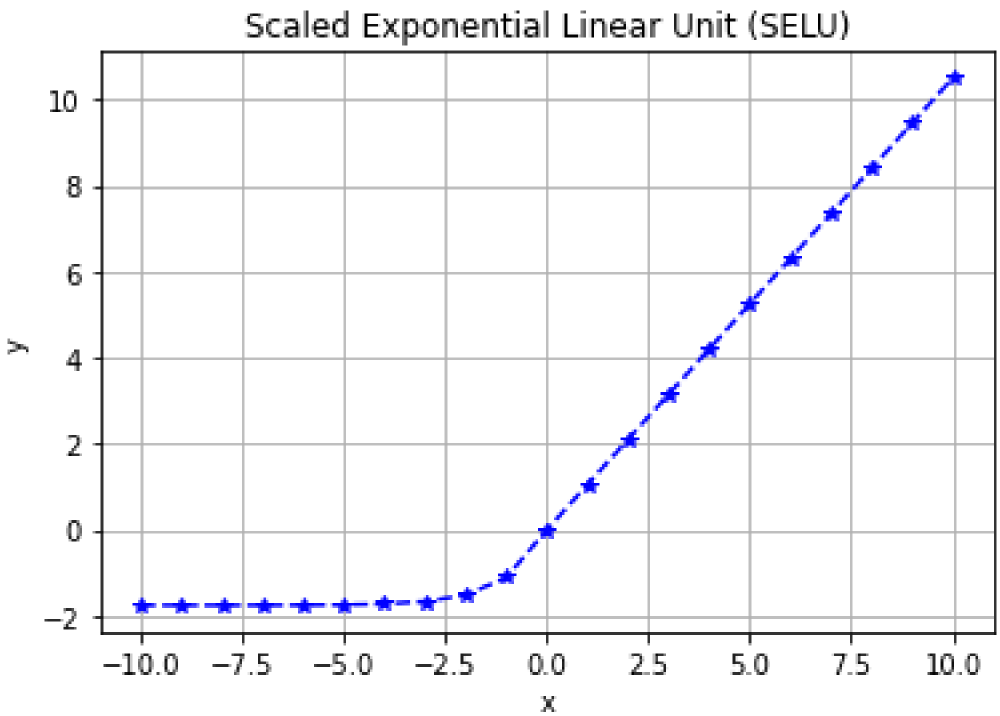

Figure 6: SELU function Graph

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.