Name: Methila Farjana
ID:19-39692-1
Section: A

Activation Function

1. Step Function:

Introduction: The step Function is one of the simplest activation Functions used in neural networks. It is a binary Function that returns a 1 if the input is greater than or equal to zero, and 0 otherwise.
Formula: f(x) = 1 if x >= 0; 0 otherwise
Advantages:

- It is computationally efficient and easy to implement.
- It can be useful in some binary classification problems.

Disadvantages:

- The Function is not continuous, which can create problems during gradient descent optimization.
- It can cause the model to get stuck in local minima during training.

2. Sigmoid Function:

Introduction: The sigmoid Function is a smooth and bounded activation Function that maps any input to a value between 0 and 1. It is commonly used in neural networks for binary classification problems.
Formula: f(x) = 1 / (1 + e^(-x))
Advantages:

- It produces a smooth output, which makes it easier to compute gradients during backpropagation.
- It is useful in binary classification problems where the output should be a probability value.

Disadvantages:

- It suffers from the vanishing gradient problem, which can slow down or even halt the learning process during training.
- It is not suitable for multi-class classification problems.

3. Tanh Function:

Introduction: The hyperbolic tangent (tanh) Function is similar to the sigmoid Function, but it maps any input to a value between -1 and 1. It is commonly used in neural networks for classification problems.
Formula: $f(x) = (e^{(x)} - e^{(-x)}) / (e^{(x)} + e^{(-x)})$
Advantages:
• It produces a smooth output, which makes it easier to compute gradients during backpropagation.

• It is useful in classification problems where the output should be a value between -1 and 1.

Disadvantages:
• It also suffers from the vanishing gradient problem, which can slow down or even halt the learning process during training.

• It is not suitable for multi-class classification problems.

4. ReLU Function:

Introduction: The rectified linear unit (ReLU) is a simple activation Function that returns the input if it is positive, and 0 otherwise. It is one of the most popular activation Functions in deep learning due to its simplicity and effectiveness.
Formula: $f(x) = \max(0, x)$
Advantages:
• It is computationally efficient and easy to implement.

• It does not suffer from the vanishing gradient problem, which makes it suitable for deep neural networks.

• It has been shown to perform well in a variety of neural network architectures.

Disadvantages:
• It can cause a problem known as "dying ReLU" during training, where some neurons become inactive and stop learning.

5. ELU Function:

Introduction: The exponential linear unit (ELU) is similar to ReLU, but it has a non-zero output for negative input values. It is designed to improve the learning speed and stability of deep neural networks.
Formula: $f(x) = x$ if $x >= 0$; $alpha * (e^{(x)} - 1)$ if $x < 0$

Advantages:
• It can speed up the learning process in deep neural networks.

• It has been shown to outperform ReLU in some cases.

• It does not suffer from the dying ReLU problem.

Disadvantages:
• It is computationally more expensive than ReLU.

6. SELU Function:

Introduction: The scaled exponential linear unit (selu) is a self-normalizing variant of the elu Function, which has been shown to improve the performance of deep neural networks.
Formula: f(x) = lambda * (e^x - 1), x <= 0
f(x) = lambda * x, x > 0
where lambda and alpha are constants that ensure the mean and variance of the output of each layer remain the same during training.
Advantages:
• The SELU Function is self-normalizing, meaning that it can maintain a stable mean and variance of activations throughout the network.

• It does not suffer from the vanishing gradient problem.

Disadvantages:
• The SELU Function can suffer from the exploding gradient problem