**NAME :** SARKAR,MD.SHARIF

**ID :** 20-43063-1

**SEC :**  A

Report on Activation function

In neural networks, activation functions introduce non - linearity, which enables the learning of complex functions. In this report, I will go over six different activation functions, their mathematical formulas, advantages, and disadvantages.

1. **Step Function**

The step function is a simple activation function that converts any input greater than or equal to zero to one and any input less than zero to zero. The step function has the following mathematical formula:

f(x) = 1, if x >= 0

f(x) = 0, if x < 0

**Advantages :**

• Efficient computation and simplicity.

• Can function as a binary classifier.

**Disadvantages :**

• not differentiable at x = 0, it cannot be used in backpropagation-based learning techniques.

• Gradient updates are a challenge to optimize because they are constant and dependent of the input.

1. **Sigmoid Function**

A well-liked activation function is the sigmoid function, which converts any real-valued input to a number between 0 and 1. The mathematical formula for the sigmoid function is as follows :

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

**Advantages :**

• Smooth and differentiable

• Can be used to model probabilities, making it useful for binary classification problems

• Well-suited for use in shallow neural networks with few hidden layers

**Disadvantages: :**

•Prone to the vanishing gradient problem, making it difficult to train deep neural networks

•Outputs are not zero-centered, which can slow down convergence during training

Similar to the sigmoid function, the tanh function converts input into a number between -1 and 1. The tanh function has the following mathematical formula :

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

1. **Tanh Function**

**Advantages :**

• Smooth and differentiable.

• More efficient than the sigmoid function for deep neural network training since outputs are zero-centered.

• Both shallow and deep neural networks can benefit from using this.

**Disadvantages :**

• Vulnerable to the vanishing gradient issue, which makes it challenging to train very deep neural networks.

1. **ReLu Function**

A popular activation function, the Rectified Linear Unit (ReLU) function transfers every negative input to zero and any positive input to itself. The ReLU function has the following mathematical formula:

f(x) = max (0, x)

**Advantages :**

•Computationally effective,

•Has the ability to speed up convergence during training,

•Immune to the vanishing gradient issue.

**Disadvantages :**

•Not differentiable at x = 0 can be problematic for backpropagation.

•Susceptible to the "dying ReLU" problem, in which a significant component of the network can cease responding and learning.

1. **PReLU) function**

The slope of the negative half of the function can be learned during training using the Parametric Rectified Linear Unit (PReLU) function, which is a version of the ReLU function. The PReLU function has the following mathematical formula:

f(x) = max (0, x) + alpha \* min (0, x).

**Advantages :**

•Can solve the "dying ReLU" issue.

•Can learn during training the slope of the negative portion of the function, improving performance.

**Disadvantages :**

•May not always enhance performance over the ReLU function.

•Costlier to compute than the ReLU function.

1. **EReLU function**

Another ReLU variant that uses the exponential function to smooth out the negative portion of the function is the Exponential Linear Unit (ELU) function. The EReLU function has the following mathematical formula:

f(x) = x, if x > 0; f(x) = alpha \* (exp(x) -1);

if x = 0;

where alpha is a hyperparameter that regulates the strength of the negative slope.

**Advantages :**

•By preventing the gradient from turning negative or zero for negative inputs, it solves the "dying ReLU" issue and speeds up training.

•Smooth and differentiable, which supports gradient-based learning and optimization.

• Zero-centered outputs can enhance learning performance.

**Disadvantages :**

• Requires more tuning of the hyperparameters (alpha) and is computationally more expensive than the ReLU function and other variants like the PReLU function.

•Limited community support and fewer resources for implementation

•optimization due to not being as popular or frequently used as other activation functions.