Name: Mehadi Hasan Reaz

ID: 20-42841-1

Section: A

Report on the Activation function

Activation functions play a crucial role in neural networks by introducing nonlinearity, which enables them to learn complex functions. In the following report, I will examine six distinct activation functions and provide information about their mathematical expressions as well as their strengths and limitations.

1. Step Function:

The step function is a simple activation function that maps all inputs greater than or equal to zero to one and all inputs less than zero to zero. The mathematical formula for the step function is as follows:

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

$f(x) = 0$, if $x < 0$

Advantages:

- Simple function with a distinct threshold value
- Useful for cases requiring binary decisions
- Computationally effective and simple to use
- Can be used for certain kinds of neural networks

Disadvantages:

- Discontinuous function, which can make it difficult to use in some optimization algorithms
- May cause issues with vanishing gradients and slow down the learning process in deep neural networks

2. Sigmoid Function:

The sigmoid function is a popular activation function that maps any real-valued input to a value between 0 and 1. The mathematical formula for the sigmoid function is as follows:

$f(x) = 1 / (1 + e^{(-x)})$

Advantages:

- Well-defined output range
- Can be useful for feature scaling
- Simple formula

Disadvantages:

- Vanishing gradient problem
- Output range limitation

3. Tanh Function:

The tanh function is similar to the sigmoid function but maps input to a value between -1 and

1. The mathematical formula for the tanh function is as follows:
f(x) = (e^x - e^(-x)) / (e^x + e^(-x))
Advantages:

- Continuous function: the tanh function is a smooth, continuous function, making it easier to optimize in many cases.
- Widely used: the tanh function has been used as an activation function in many neural network architectures.
- Can be useful for feature scaling: the tanh function can be used to scale features to a similar range, which can improve the performance of some types of neural networks.

Disadvantages:

- Output range limitation: Calculating the tanh formula's exponential function d 1 may cause saturation and limit the neural network's ability to learn complex patterns.
- Computationally expensive: Calculating the tanh formula's exponential function can be computationally expensive, which may slow down the training process.

4. Relu Function:

The Rectified Linear Unit (ReLU) function is a popular activation function that maps any negative input to zero and any positive input to itself. The mathematical formula for the ReLU function is as follows:
f(x) = max (0, x)
Advantages:

- Computationally efficient
- Can accelerate convergence during training
- Does not suffer from the vanishing gradient problem

Disadvantages:

- Not differentiable at x = 0, which can cause issues during backpropagation
- Prone to the "dying ReLU" problem, where a large portion of the network can become non-responsive and stop learning

5. PReLU Function:

The Parametric Rectified Linear Unit (PReLU) function is a variation of the ReLU function that allows the slope of the negative part of the function to be learned during training. The mathematical formula for the PReLU function is as follows:
f(x) = max (0, x) + alpha * min (0, x)
Advantages:

- Can address the "dying ReLU" problem
- Can learn the slope of the negative part of the function during training, leading to improved performance

Disadvantages:

- More computationally expensive than the ReLU function
- May not always improve performance over the ReLU function

6. EReLU Function:

The Exponential Linear Unit (ELU) function is another variation of the ReLU function that smooths out the negative part of the function using the exponential function. The mathematical formula for the EReLU function is as follows:

f(x) = x, if x > 0

f(x) = alpha * (exp(x) - 1), if x <= 0

where alpha is a hyperparameter that controls the magnitude of the negative slope.

Advantages:

- Addresses the "dying ReLU" problem by preventing the gradient from becoming zero or negative for negative inputs, which can speed up training
- Smooth and differentiable, which helps with optimization and gradient-based learning
- Outputs are zero-centered, which can improve learning performance

Disadvantages:

- More computationally expensive than the ReLU function and other variations like the PReLU function
- Additional hyperparameters to tune (alpha)
- Not as widely used or well-known as other activation functions, so there may be less community support and fewer resources available for implementation and optimization.