Name: Sadman Sanid Tanim

Id: 20-43866-2

Report on Activation function

Activation functions in neural networks introduce nonlinearity, allowing neural networks to learn complex functions. In this report, I will discuss six different activation functions, their mathematical formulas, advantages, and disadvantages.

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 and computationally efficient
- Can be used as a binary classifier

Disadvantages:

- Not differentiable at x = 0, making it unsuitable for use in backpropagation-based learning algorithms
- Gradient updates are constant and not dependent on the input, making it difficult to optimize.

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:

- 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

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:

- Smooth and differentiable
- Outputs are zero-centered, making it more effective for training deep neural networks than the sigmoid function
- Well-suited for use in shallow and deep neural networks Disadvantages:

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

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. <u>EReLU function:</u>

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**.

6. <u>PReLU function:</u>

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