A Report on Activation Functions

1.STEP FUNCTION

The perceptron network employs the Step activation function. This is typically employed in single-layer networks to provide binary (0 or 1) or bipolar output (-1 or 1). These are known respectively as Binary Step Function and Bipolar Step Function. Here, a function's output is 1 (a neuron will fire) if the input value exceeds a threshold value; otherwise, it is 0 or -1 in the case of a bipolar step function (neuron will not fire).

When an input value crosses a threshold, the step function's output quickly changes from 0 to 1 and is discontinuous. The step function is less suitable for training neural networks using gradient-based optimization methods since the derivative is usually always 0. As a result, updating the network's weights during backpropagation is difficult.

Advantages:

I. Computationally efficient and simple.
II. Can be used as a binary classifier.

Disadvantages:

I. It is not appropriate for use in backpropagation-based learning algorithms because it is not differentiable at x = 0.
II. Gradient updates are a challenge to improve because they are constant and independent of the input.

2.SIGMOID FUNCTION

This activation function associates a stream of input data with a range (0, 1). Sigmoid, in contrast to the step function, outputs a range of values instead of just 0 or 1. The sigmoid function has some drawbacks despite being superior to the ones previously discussed and having its uses (particularly in tasks like binary classification). These drawbacks include the fact that very large and very small input values can interfere with backpropagation because these saturated neurons "kill" the gradients. The output of the sigmoid function is not 0-centered because the range is (0, 1), which is another disadvantage that affects backpropagation. Finally, exponential functions are computationally expensive, which can cause the network to lag.

Advantages:

I. Well-suited for use in shallow neural networks with few hidden layers.
II. Smooth and different.
III. It can be applied to probabilistic modeling, which makes it helpful for binary classification issues.

Disadvantages:

I. Outputs are not zero-cantered, which can slow down convergence during training.

II. Vulnerable to the vanishing gradient issue, which makes deep learning challenging.

3.TANH FUNCTION

This activation function shares some similarities with the sigmoid in that it similarly converts the input values into an s-shaped curve, however in contrast to the sigmoid, it has a range of (-1, 1) and a center of zero, which addresses one of its drawbacks. Tanh stands for the hyperbolic tangent, which, like the regular tangent, is simply the hyperbolic sine divided by the hyperbolic cosine.

Tanh is an exponential function, therefore while it can be more efficient than the sigmoid, it still has the same challenges with backpropagation when dealing with extremely big or extremely small values as the sigmoid.

Advantages:

I. Smooth and unique.
II. Compared to the sigmoid function, the outputs are zero-centered, making it more useful for teaching deep neural networks.
III. Well-suited for use in shallow and deep neural networks.

Disadvantages:

I. Vulnerable to the disappearing gradient issue, which makes training very deep neural networks challenging

4.RELU - RECTIFIED LINEAR UNITS

This activation method is more recent and popular. Rectified Linear Unit is what it's called, and it looks like this. ReLU's simplicity is one of its many appealing qualities. As one can see, all it does is leave positive values alone and substitute negative values with 0. This is substantially faster computationally and eliminates the issue of "killing" the gradients of large and small values. Moreover, ReLU-based networks typically converge roughly six times quicker than sigmoid- and tanh-based networks in practice. ReLU still has some issues, though. The first issue is that it isn't 0-centered, which might be problematic for training. But more crucially, it doesn't really deal with negative inputs in a meaningful way.

Advantages:

I. It is a simple function that is computationally efficient to compute.
II. It does not suffer from the problem of vanishing gradients for large input values, making it easier to train deep neural networks.
III. It can help to sparsity the activation of the network, by setting some of the activations to zero, which can help to prevent overfitting.

Disadvantages:

I. Not differentiable at x = 0, which can cause issues during backpropagation.
II. vulnerable to the "dying ReLU" problem, in which a significant portion of the network can cease responding and learning.

5.ELU - EXPONENTIAL LINEAR UNIT

The Exponential Linear Unit (ELU) is a neural network activation function. ELUs, in contrast to ReLUs, have negative values, allowing them to, with less computational cost than batch normalization, move mean unit activations closer to zero. Due to a diminished bias shift effect, mean shifts toward zero accelerate learning by bringing the normal gradient closer to the unit natural gradient. Even while LReLUs and PReLUs can be negative, they cannot guarantee a deactivation state that is resistant to noise. With smaller inputs, ELUs saturate to a negative value, which reduces the forward propagated variance and information.

Advantages:

I. The ELU function is smooth everywhere, which can help optimization algorithms converge faster.
II. Faster convergence than Relu
III. Unlike ReLU, which outputs 0 for negative inputs, the ELU function can output negative values, which can be beneficial for some types of data.

Disadvantages:

I. The ELU function requires the use of an exponential function, which can be computationally expensive.
II. The ELU function is not as widely used as other activation functions like ReLU, which means that there is less information available on how to tune its hyperparameters.
III. The ELU function can lead to overfitting if the alpha parameter is not properly tuned.

6. SELU - SCALED EXPONENTIAL LINEAR UNIT

SELUs, or Scaled Exponential Linear Units, are activation functions that induce self-normalization. SELU network neuronal activations automatically converge to a zero mean and unit variance. If x is larger than 0, the output result is x multiplied by lambda. If the input value x is less than or equal to zero, we have a function that goes up to 0, which is our output y, when x is zero. Essentially, when x is smaller than zero, we take the exponential of the x-value minus 1, then we multiply it with alpha α and lambda λ.

Advantages:

I. Designed to achieve self-normalization in deep neural networks.
II. Leading to faster convergence and better performance.
III. Zero-centered and has a smooth curve.

Disadvantages:

I. Requires proper initialization of weights and biases to achieve self-normalization.