Name: Shahriar Siddique Ayon ID: 20-42265-1

A Report on Activation Function

1.STEP FUNCTION

The Step activation function is used by the perceptron network. To provide binary (0 or 1) or bidirectional output, this is usually used in singlelayer networks. (-1 or 1). They are referred to as Binary Step Function and Bipolar Step Function, respectively. Here, if the input value surpasses a threshold value, a function's output is 1 (a neuron will activate); otherwise, it is 0 or -1 in the case of a bipolar step function. (neuron will not fire).

The output of the step function abruptly changes from 0 to 1 when an input number exceeds a threshold. Since the derivative is typically always 0, the step function is less appropriate for training neural networks using gradient-based optimization techniques. As a consequence, it is challenging to update the network's weights during backpropagation.

Advantages:

i. Can be used as a binary classifier.
ii. Computationally efficient and simple.

Disadvantages:

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

2.SIGMOID FUNCTION

With the help of this activation function, a trickle of incoming data is linked to a range. (0, 1). In opposition to the step function, the sigmoid function produces a range of values rather than just 0 or 1. Despite being better to the ones previously discussed and having applications, the sigmoid function has some disadvantages. (particularly in tasks like binary classification). Due to these overloaded neurons' ability to "kill" the gradients, extremely high and extremely low input values can conflict with backpropagation. Another drawback to backpropagation is that the sigmoid function's result is not 0-centered because its range is (0, 1). Finally, because exponential functions require costly processing, the network may lag.

Advantage:

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

Disadvantage:

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 resembles the sigmoid in that it generates an s-shaped curve from the input values, but unlike the sigmoid, it has a range of (-1, 1) and a center of zero, addressing one of its shortcomings. Tanh stands for the hyperbolic tangent, which is just the hyperbolic sine split by the hyperbolic cosine, like the normal tangent.

Tanh is an exponential function, so even though it has the potential to be more effective than the sigmoid, it still faces the same difficulties with backpropagation when handling extremely large or extremely tiny numbers as the sigmoid.

Advantage:

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

Disadvantage:

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. Its name is Rectified Linear Unit, and it has the following appearance. One of ReLU's many attractive features is how straightforward it is. As can be seen, it merely replaces negative numbers with 0 and leaves positive values alone. The gradients of big and small numbers are no longer "killed," and this is computationally much quicker. Furthermore, in reality, ReLU-based networks usually converge six times faster than sigmoid- and tanh-based networks. However, ReLU still has some problems. It is not 0-centered, which may be a difficulty for training, is the first concern. But more importantly, it doesn't really respond meaningfully to negative stimuli.

Advantage:

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

Disadvantage:

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

5.ELU - EXPONENTIAL LINEAR UNIT

An activation function for neural networks is called an exponential linear unit (ELU). ELUs, as opposed to ReLUs, have negative values, which enables them to reduce mean unit activations' distance from zero with less processing expense than group normalization. Mean shifts toward zero speed learning by bringing the normal gradient closer to the unit natural gradient due to a reduced bias shift effect. LReLUs and PReLUs have the potential to be negative, but they cannot ensure a noise-resistant shutdown condition. ELUs saturate to a negative number at lower input sizes, which lessens the forward transmitted variance and information.

Advantage:

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

Disadvantage:

i. The ELU function requires the use of an exponential function, which can be computationally expensive.
ii. The ELU function can lead to overfitting if the alpha parameter is not properly tuned.
iii. 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.

6. SELU - SCALED EXPONENTIAL LINEAR UNIT

The activation functions known as SELUs, or Scaled Exponential Linear Units, cause self-normalization. Neuronal activations in the SELU network naturally coalesce to zero mean and unit variance. The final outcome is x multiplied by lambda if x is greater than 0. We have a function that moves up to 0, which is our output y, when the input number x is less than or equivalent to 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.

Advantage:

i. Leading to faster convergence and better performance.
ii. Designed to achieve self-normalization in deep neural networks.
iii. Zero-centered and has a smooth curve.

Disadvantage:

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