American International University Bangladesh

MID ASSIGNMENT - 2 Report

Course: Computer Vision & Pattern Recognition

Name: Zishan Ahmed

ID: 20-42085-1

Section: **A**

1. Sigmoid

A Sigmoid function is a mathematical function whose characteristic curve is in the shape of a S. This function takes any value as input and gives output between 0 and 1. Sigmoid function can be represented as followed:

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$$

Advantages:

I. It has a smooth gradient.
II. It is essential in logistic regression model or where probabilities are predicted.

Disadvantages:

I. The output of the activation function will always be in the range (0,1), while the output of the linear function will always be in the range (-inf, inf).So it's limited to a certain range.
II. At the end of the sigmoid function, the values of Y vary very little when the value of X variations, which gives the problem of vanishing gradient.
III. The output isn't zero centered.

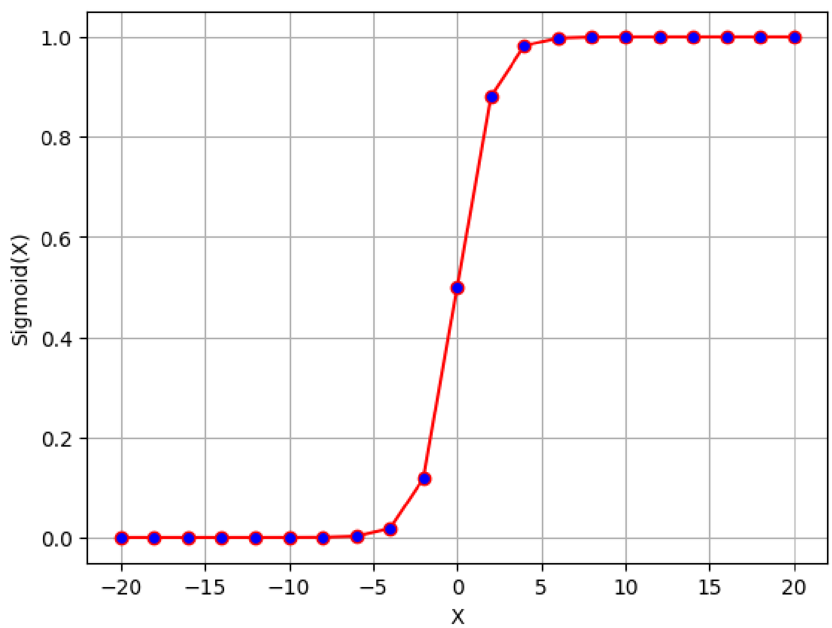

Figure 1: Sigmoid Graph

2. Tanh

The Tan(h), Tan(H), and tanh(h) functions all relate to the same thing: the hyperbolic tangent activation function. Its shape, an S, is highly reminiscent of the sigmoid activation function. Any real number can be fed into the function, and it will return a number between -1 and 1. The output value will be closer to 1 for bigger or positive inputs and closer to -1 for smaller or negative inputs. Tanh function can be represented as followed:

$$
\begin{aligned}
f(x) &= \tanh(x) \\
&= \frac{e^x - e^{-x}}{e^x + e^{-x}}
\end{aligned}
$$

<u>Advantages:</u>

I. The outputs are zero centered.
II. In order to normalize the outputs of neurons in a neural network and avoid the vanishing gradients problem, the tanh function can be used because its output is always in the range [-1, 1].
.

<u>Disadvantages:</u>

I. The tanh function has a slow convergence problem and the vanishing gradients problem since it is saturated at both ends of the range.
II. When using the tanh function as the activation function, it can be more challenging to interpret the results of the neural network compared to when using the sigmoid function.

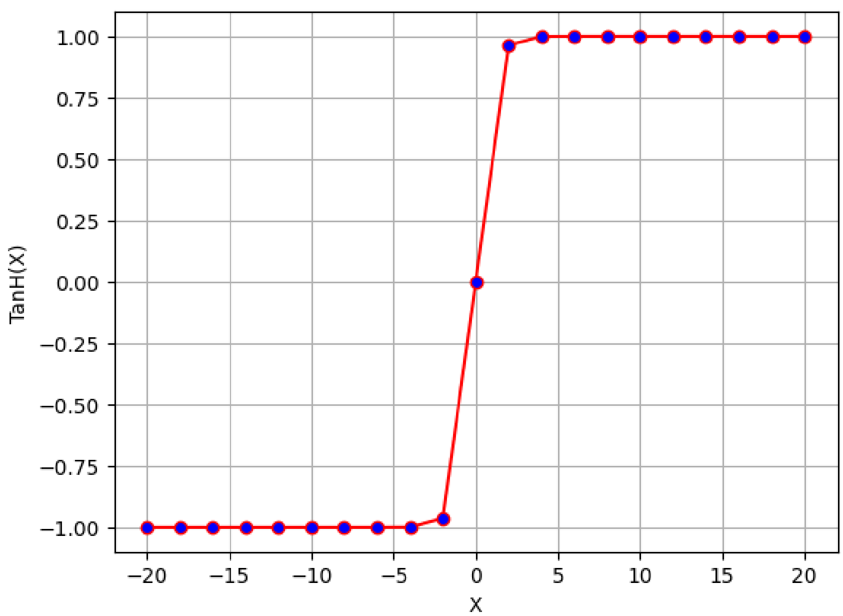

Figure 2: Tanh Graph

3. Step

One popular type of activation function in neural networks is the step function. It compares the input to a predetermined threshold and returns a binary result of 0 or 1 as output. It can be represented as followed:

$$f(x) = \begin{cases} 1 & \text{if } x > \theta \\ 0 & \text{if } x \leq \theta \end{cases}$$

Advantages:

I. The step function is a straightforward one-parameter function that takes a threshold value as input. It's straightforward to implement in most languages.
II. For data with distinct positive and negative exceptions, the step function may be an useful method.
III. The step function produces a binary output, which can be useful for certain tasks such as binary classification problems.

Disadvantages:

I. It can be challenging to apply optimization algorithms that rely on gradients with the step function since it is not differentiable.
II. For applications that need more detailed predictions, the step function's binary output may be insufficient.
III. The step function can experience saturation, which results in very little or no learning because the gradient is so small.

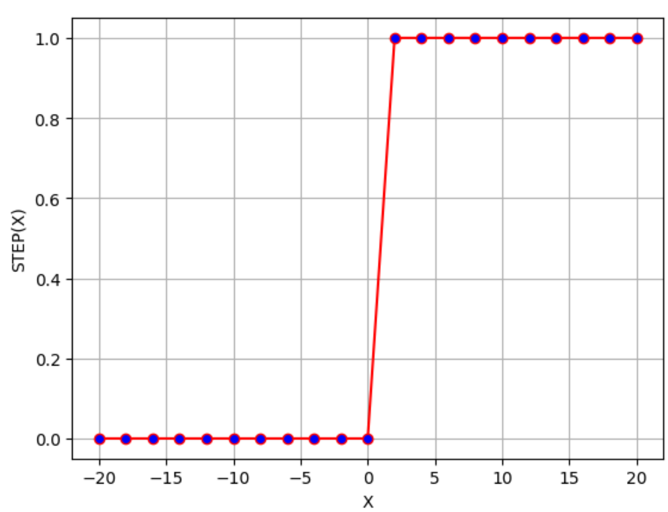

Figure 3: Step function graph

4. ReLU

A Rectified Linear Unit, or ReLU, is a type of linear measurement. It appears and behaves like a linear algorithm, but it actually gives the same benefits as Sigmoid while being more efficient. To put it another way, ReLU can go anywhere from 0 to infinity. The concept can be shown in the following way,

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

Advantages:

I. In contrast to more complex activation functions like sigmoid and tanh, the ReLU activation function uses minimal computation.
II. The issue of a vanishing gradient is bypassed and fixed.
III. Sparsity in the activation outputs of a neural network is a potential benefit of the ReLU activation function for preventing overfitting and increasing generalization.

Disadvantages:

I. Being a non-linear function, the ReLU activation function is not smooth.
II. An unlimited number of outputs is possible from the ReLU activation function. When the training rate is fast, this might lead to numerical instability.
III. Since the gradient will be zero for activations in the region (x<0) of ReLU, the weights will not be updated as the network descends. Neurons entering this condition will no longer respond to changes in error since the gradient is 0. This issue is called the dead neuron problem.

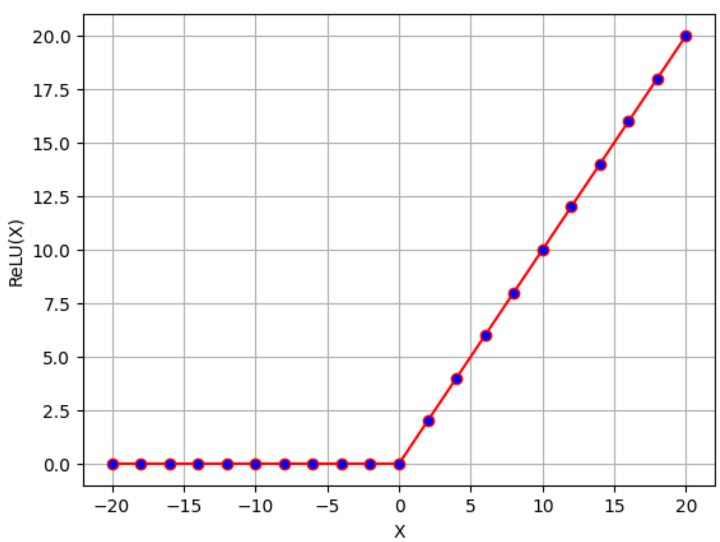

Figure 4: ReLU graph

5. Leaky ReLU

The activation function known as Leaky ReLU is a variation of the Rectified Linear Unit (ReLU). The "dead neuron" issue that might arise with the traditional ReLU function is tackled by this variant. The function is slightly modified from ReLU, which is as followed:

$$f(x) = max\ (0.1x, x)$$

Advantages:

I. By returning non-zero values for negative inputs, Leaky ReLU prevents the dead-neuron problem from occurring and aids in the learning process by keeping the gradient during backpropagation from going to zero.
II. For larger datasets and more sophisticated designs, Leaky ReLU has been proven to perform better than ReLU.

Disadvantages:

I. Because of its increased complexity, evaluating Leaky ReLU may demand more processing power than evaluating ReLU.
II. It is important to select a suitable value for the leakiness hyperparameter, which is the slope of the negative component of the function.
III. Leaky ReLU may outperform ReLU on some datasets and architectures, however this is not always the case.

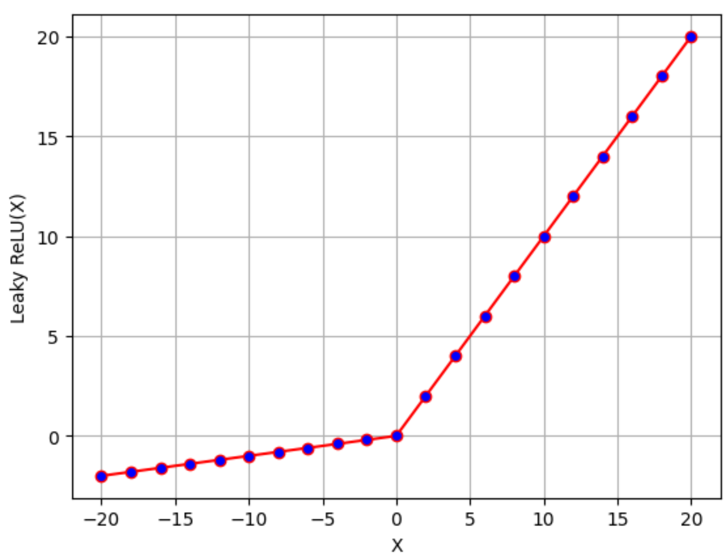

Figure 5: Leaky ReLU graph

6. ELU

The Exponential Linear Unit (ELU) is an alternative activation function to the Rectified Linear Unit (ReLU) that attempts to address some of its drawbacks. On positive inputs, an ELU activation layer does an identical operation, but on negative inputs, it applies an exponential nonlinearity. It can be represented as followed:

$$f(x) = \begin{cases} 0 & if\ x > 0 \\ \alpha(\exp(x) - 1) & if\ x \leq 0 \end{cases}$$

Advantages:

I. The ELU function is continuously differentiable, whereas the ReLU activation function is only differentiable up to x = 0.
II. The sparsity restriction induced by the ELU function can be used to assist mitigate overfitting in neural networks.
III. When compared to other activation functions like the hyperbolic tangent and the sigmoid, the ELU function is less likely to suffer from the problem of negative saturation.

Disadvantages:

I. It takes more processing power to evaluate the ELU function than other activation functions like ReLU since it requires the evaluation of exponentials.
II. The amount of negative saturation is controlled by the hyperparameter alpha of the ELU function. The model's effectiveness can be affected by the choice of alpha (α).

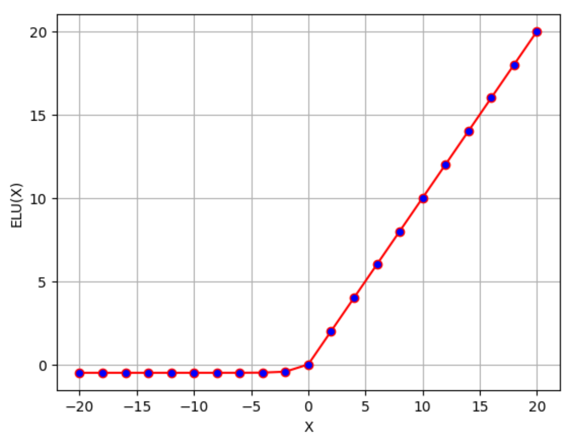

Figure 6: ELU graph

7. SELU

For better convergence and overall performance in deep neural networks, the Scaled Exponential Linear Unit (SELU) activation function has become increasingly popular in recent years. Neuronal activations in a SELU network ultimately converge to a normal distribution with a mean of zero and a variance of one. The following is a representation of it:

$$f(x) = \lambda x \qquad \text{if } x > 0$$

$$f(x) = \lambda\alpha(e^{x} - 1) \qquad \text{if } x \leq 0$$

Advantages:

I. As a self-normalizing function, the SELU can be used to minimize the vanishing/exploding gradient issue that sometimes arises in deep neural networks. Since this is the case, training can converge more quickly and reliably.
II. It has been demonstrated that the SELU function is superior to other activation functions, such as the ReLU, ELU, and sigmoid functions, for a wide range of tasks when using deep neural networks.
III. The SELU function can be used to enhance neural network generalization by creating a noise-like feature that works to reduce overfitting.
IV. Unlike ReLU and ELU, which typically require batch normalizing for optimal performance, the SELU function can function well even when batch normalization is not applied.

Disadvantages:

I. In comparison to simpler activation functions like ReLU and ELU, the evaluation of the SELU function can be time-consuming and resource-intensive due to the need to calculate exponential and square root functions.
II. There are various requirements for the SELU function to be self-normalizing, such as a specific approach for initializing the weights and standardizing the input to have zero mean and unit variance.

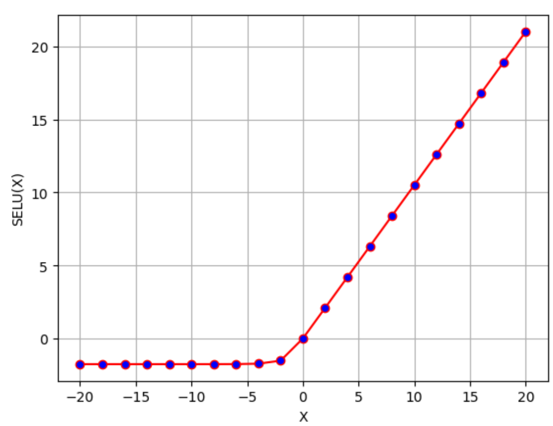

Figure 7: SELU graph