**Report On Activation Functions**

**Introduction:**

Neural networks require activation functions, which are utilized to give the model non-linearity and decision-making skills. The activation function receives the weighted total of inputs, which then performs a nonlinear transformation to produce the neuron's output. Some frequently used activation functions and their characteristics will be covered in this study. In assignment-2, we discuss the Classification of activation functions. We discussed the step function, sigmoid function, Tanh function, ReLU function, Elu function, and SeLU function. In this report, we are going to make a comparison between those functions and try to elaborate on their graphical output.

At first, the Step function is a binary function that outputs a value of 1 if the input is greater than or equal to a threshold value and 0 otherwise. On the other side, the sigmoid function is a smooth function that maps any input to a value between 0 and 1. The step function has a discontinuous graph and is not differentiable at the threshold point. However, the sigmoid function has a continuous and differentiable graph, making it suitable for optimization algorithms such as gradient descent. The step function is mainly used in binary classification problems, while the sigmoid function is used in logistic regression and neural network models. After this short comparison, we can see that the sigmoid function is better than the step function. However, there are many conditions where the step function will be the best choice against the sigmoid function.

We know the advantage and disadvantages of the Tanh function. What about a comparison between Tanh and sigmoid function?

Sigmoid and tanh functions share some similarities but differ in their range, shape, symmetry, and applicability. For example, the sigmoid function maps any input to a value between 0 and 1, while the tanh function maps any input to a value between -1 and 1. The tanh function has a "tanh"-shaped curve that is symmetrical around the origin, whereas the sigmoid function has an "S"-shaped curve. In the vicinity of the origin, the tanh function exhibits steeper gradients than the sigmoid function. The sigmoid function is not symmetric around the origin, while the tanh function is symmetric. This property of the tanh function allows for negative and positive inputs.

Summarizing, it can be said that while sigmoid and tanh functions have certain similarities, they also have differences in symmetry, range, and application. The exact situation at hand and the properties of the data will determine whether to use sigmoid or tanh functions.

Now, we will describe the similarities and differences between Tanh and ReLU functions.

Tanh (hyperbolic tangent) and ReLU (Rectified Linear Unit) are popular activation functions in neural networks. The tanh function maps any input to a value between -1 and 1, while the ReLU function maps any input less than or equal to zero, and any input greater than zero to the input value itself. Therefore, the range of the ReLU function is wider than that of the tanh function. We know tanh function has a "tanh"-shaped curve that is symmetric around the origin, while the ReLU function is a simple linear function for inputs greater than zero and is zero for inputs less than or equal to zero. Tanh function has a vanishing gradient problem, while the ReLU function is considered more efficient and does not suffer from this problem. ReLU function can suffer from the "dead neurons" problem, where some neurons can become permanently inactive and produce zero outputs. This can happen when the input value to the ReLU function is always less than zero.

In the end, the Tanh function may be more suitable for feedforward neural networks, while the ReLU function may be more suitable for deep learning architectures such as CNNs and RNNs.

According to ELU function maps, any input less than or equal to zero equals an exponential function of the input, and any input greater than zero equals zero to the input value itself. ELU function is similar to the ReLU function for inputs greater than zero but has a smooth curve for inputs less than or equal to zero. Most importantly, the ReLU function is not continuous at zero, while the ELU function is continuous at all points. The ReLU function is computationally more efficient than the ELU function because the ELU function requires the computation of exponential functions. The ELU function has been shown to perform better than the ReLU function in some cases, especially in tasks that require faster convergence and lower error rates.

Lastly, we have Elu and Selu to compare with each other.

Below here, we are going to discuss it.

Both ELU and SELU functions have a similar shape for inputs greater than zero, but SELU has a specific shape for inputs less than or equal to zero, a scaled exponential function. Moreover, the common thing between them is that both functions are continuous and differentiable at all points. In the vanishing gradient issue, SELU has been shown to perform better than ELU in avoiding the problem. The normalization term in the SELU function indicates that the function's output is normalized with the input distribution, leading to quicker convergence and improved generalization. SELU function requires additional normalization terms, whereas ELU only requires the computation of exponential functions.

From the above discussion, the SELU function may be more suitable for deep neural networks due to its normalization term and better performance, especially for tasks that require high accuracy. At the same time, the ELU function can also be used in deep neural networks but may not be as effective as SELU in certain cases.

**Conclusion:**

In conclusion, activation functions are an essential component of neural networks, and each activation function has unique properties and advantages. The choice of activation function depends on the problem at hand, and it is essential to select the right activation function to ensure better performance and faster convergence.