Step function:

- Advantage: It is simple and easy to understand. It is also computationally efficient.
- Limitation: It is not differentiable and can cause issues with gradient-based optimization algorithms. It also only allows for binary outputs, which is not suitable for more complex tasks.

Sigmoid function:

- Advantage: It produces a smooth and continuous output that is suitable for gradient-based optimization algorithms. It can also be interpreted as a probability distribution useful in certain applications.
- Limitation: It suffers from the vanishing gradient problem, which can cause issues with deep neural networks. It also has a limited range (0 to 1), which can lead to saturation and hinder the learning process.

TanH function:

- Advantage: It has a wider range than the sigmoid function (-1 to 1), which can be useful in some applications. It also produces a smooth and continuous output that is suitable for gradient-based optimization algorithms.
- Limitation: It also suffers from the vanishing gradient problem, which can cause issues with deep neural networks. It can also cause a problem with bias if the data is not centered around zero.

ReLU function:

- Advantage: It is computationally efficient and allows for faster training of deep neural networks. It also addresses the vanishing gradient problem and can lead to better performance in certain applications.
- Limitation: It suffers from the dying ReLU problem, where some neurons can become inactive and stop learning. It also produces a non-differentiable output for the negative values of x.

ELU function:

Advantage:

It addresses the dying ReLU problem and produces a smooth and continuous output that is suitable for gradient-based optimization algorithms. It can also produce negative outputs, which can be useful in some applications.

Limitation:

It can be computationally expensive compared to other activation functions. It also has a limited range of alpha values that can be used, which can limit its flexibility in certain applications.

Selu Function:

Advantages:

1. Self-normalization: The SELU activation function is designed to enable self-normalization in deep neural networks, which can help to mitigate the vanishing/exploding gradient problem that can occur when using other activation functions. In other words, SELU can help ensure that each layer's output in a neural network has a similar distribution of values, which can lead to better performance and faster training.
2. Improved performance: The authors of the original SELU paper found that using the SELU activation function led to improved performance on a variety of tasks, including image classification and speech recognition.
3. Sparsity: Like ReLU, the SELU activation function can help to introduce sparsity into a neural network, which can help to reduce overfitting and improve performance.

Limitations:

1. Initialization constraints: The SELU activation function relies on specific initialization constraints for the weights and biases in a neural network. If these constraints are not met, the self-normalization property of the activation function may not hold, and the performance of the network may suffer.
2. Limited applicability: The SELU activation function is designed to work with feedforward neural networks that use dense layers. It may not work as well with other types of neural networks, such as recurrent neural networks or convolutional neural networks.
3. Computationally expensive: The SELU activation function is more computationally expensive than some other activation functions, such as ReLU. This can lead to slower training times, especially for large neural networks.