AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH

RAMEEN FARHAN

20-41981-1

Course: computer vision and pattern recognition

Section : A

Activation function: an activation function defines that outputs a smaller value for tiny inputs and a higher value if its inputs are greater than a threshold. The activation function "fires" if the inputs are big enough; otherwise, nothing happens.

Step function

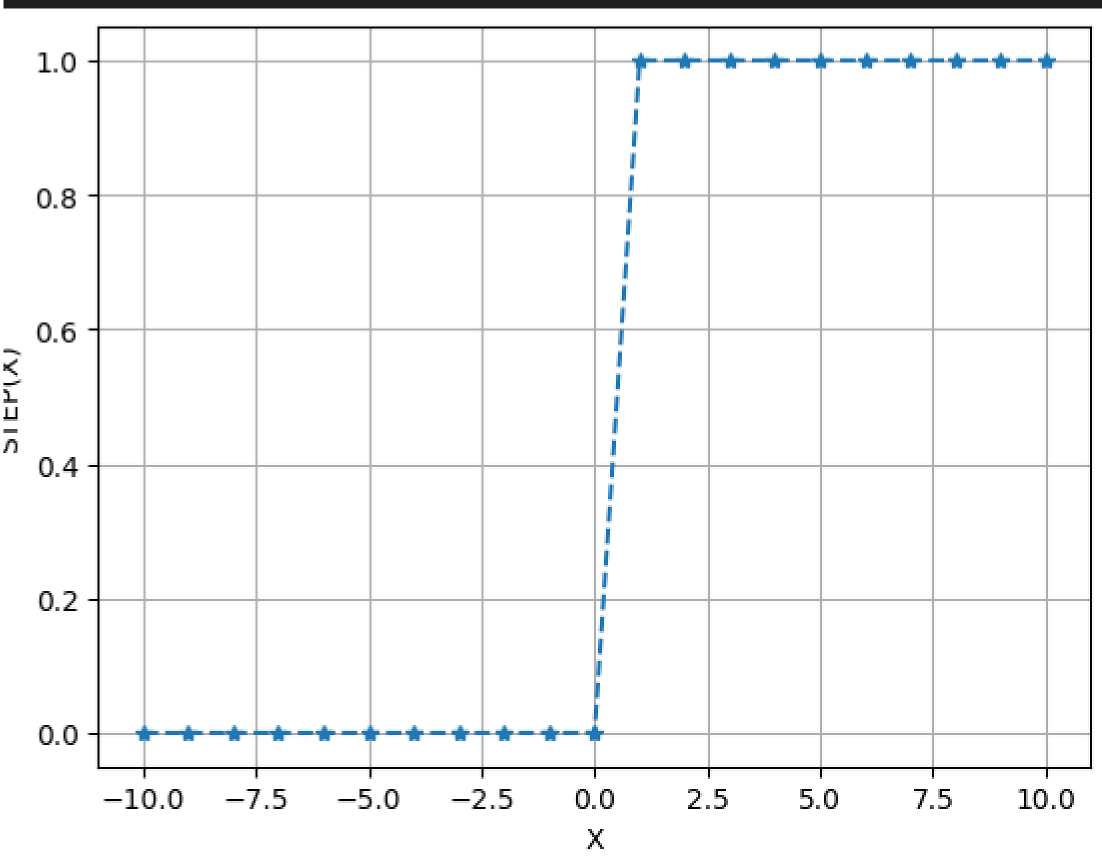
A step function, sometimes known as a staircase function, is a piecewise constant function with a limited number of parts, according to the definition given in mathematics. In other words, one can think of a function on the real numbers as a finite linear combination of indicator functions of given intervals.

Formula

If $x >= 0$, then $f(x) = 1$.

$f(x) = 0$, if $x < 0$

```python
# step function
y = list(map(lambda n: 1 if n>0.5 else 0, x))
plot_graph(y,"STEP(X)")
```



<u>Advantages</u>

1.Simplicity

2.Discrete output

3.Robustness

<u>Disadvantages</u>

1.Non-differentiability

2. Limited expressivity

3.Gradient vanishing

Sigmoid Function

A well-liked activation function is the sigmoid function, which converts any real-valued input to a number between 0 and 1.

Formula

1 / (1 + e(-x))f(x)

Advantages

• Sigmoid function outputs values in the range of 0 to 1, which makes it suitable for binary classification problems.

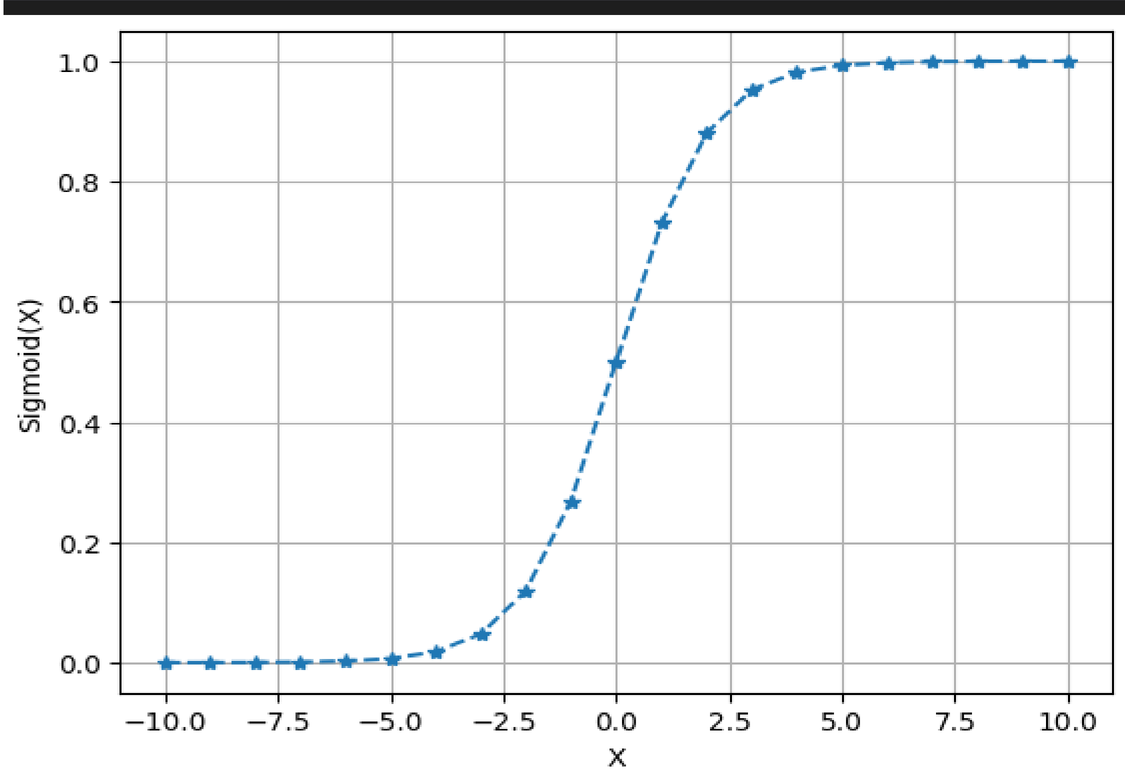
• It is a standard function and its properties are well known and understood, making it easy to use.

Disadvantages

• It is expensive compared to other activation function.

• This function is not suited for regression problems.

```
# sigmoid function
y = 1 / (1 + np.exp(-x))
plot_graph(y, "Sigmoid(X)")
```



Tanh Function

Similar to the sigmoid function, the tanh function converts input into a number between -1 and 1.

Formula

f(x) is calculated as (ex - e(-x)) / (ex + e(-x)).

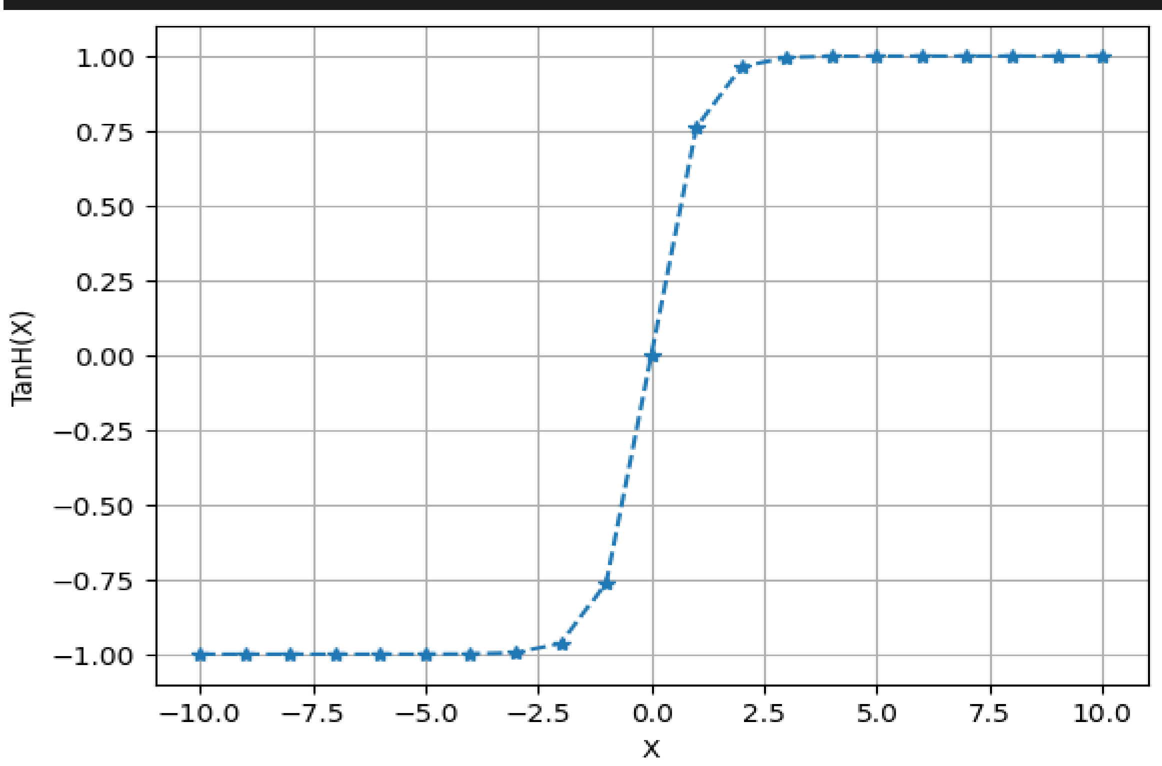
Advantages

- It can introduce non-linearity into the neural network, enabling it to learn complex patterns in the data.
- Tanh is symmetric around the origin, which means it can model negative inputs as well as positive inputs.

Disadvantages

The output of the tanh function is not centered around zero, which can make optimization more
challenging.

```
# tanh function
y = (np.exp(2*x) - 1) / (np.exp(2*x) + 1)
plot_graph(y, "TanH(X)")
```



Relu Function
A well-liked activation function, the Rectified Linear Unit (ReLU) function transfers every negative input to zero and any positive input to itself.

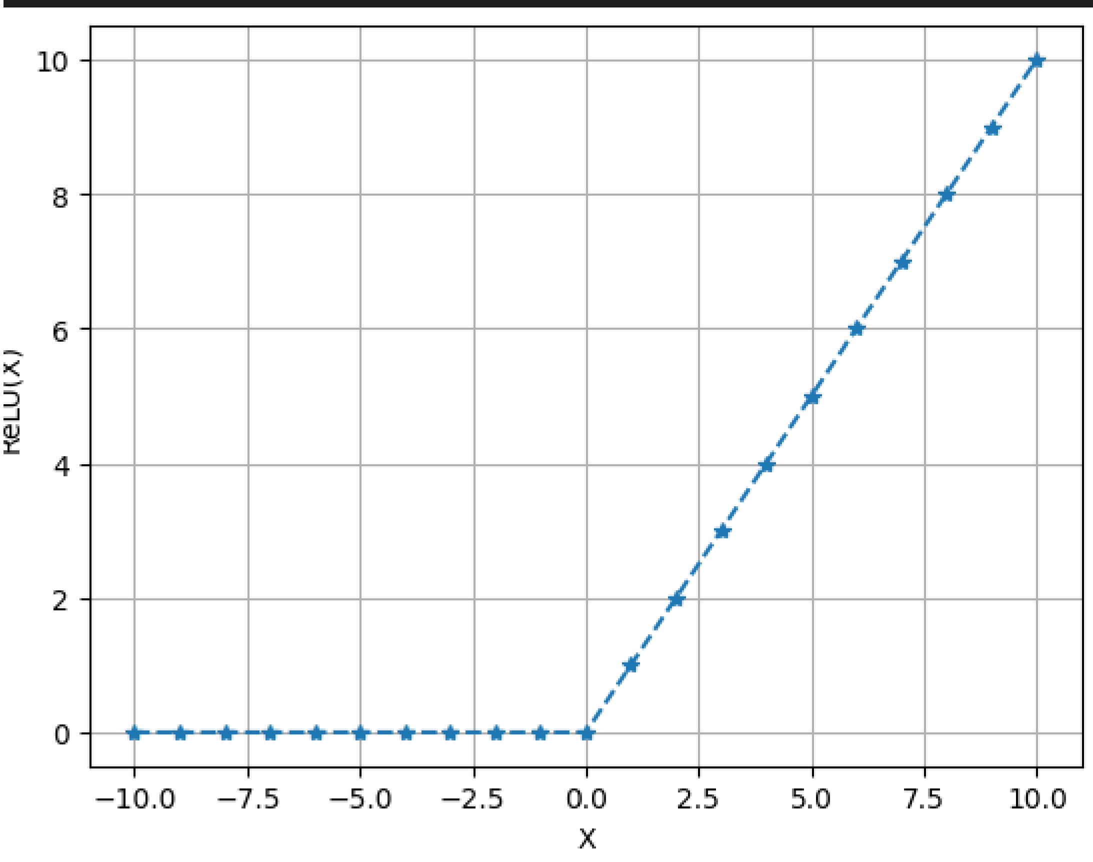
Formula
max(f(x) (0, x)
<u>Advantages</u>
- Sparsity
- Non-linearity
- Computational efficiency

<u>Disadvantages:</u>
- Unbounded output
- Not suitable for negative inputs

<u>PReLU Function</u>
An activation function known as a Parametric Rectified Linear Unit (PReLU) is a traditional rectified unit with a slope for negative values.
Formula
f(x) = max (0, x) + alpha * min (0, x)
<u>Advantages</u>
- Non-linearity
- Avoids dead neurons
- Better performance

<u>Disadvantages</u>
- Overfitting
- Computational cost
- Model complexity

```
#  ReLU Function
y = list(map(lambda a: a if a>=0 else 0, x))
plot_graph(y,"ReLU(X)")
```



<u>EReLU Function</u>

With no additional parameters and no overfitting risk, EReLU enhances model fitting.Moreover, by utilizing EReLU and parametric ReLU, we propose Elastic Parametric Rectified Linear Unit (EPReLU) (PReLU). EPReLU has the ability to enhance network performance even more.

Formula

If x > 0 then f(x) = alpha * (exp(x) - 1), otherwise f(x) = x.

<u>Advantages</u>

• Avoids the ‘dying ReLU’ problems
• Better generalization
• Parameterized

Disadvantages

• High computational cost
• Sensitive to initialization
• Limited resources.

```
# ELUs Function
def elu(x, alpha=0.01):
    return np.where(x > 0, x, alpha*(np.exp(x)-1))

plt.title('ELU Function')
plt.plot(x, elu(x), '*--b')
plt.show()
```