Importing Libraries

In [ ]:

```
from math import *
from matplotlib.pyplot import *
from numpy import *
```

Activation Function Definitions

Input is a list of x values, output is the values modified by the Activation Function

In [ ]:

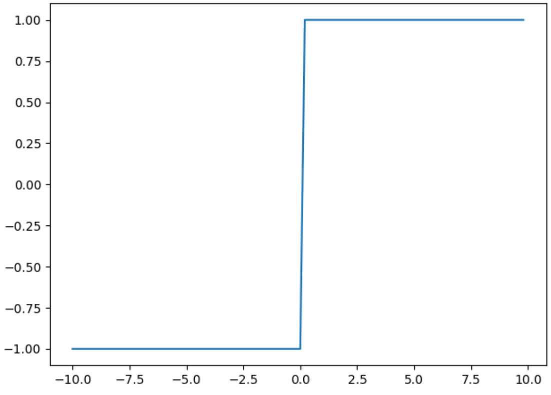
```
# Threshold is 0
def symmetric_hard_limit_transfer(x_values):
    return [1 if x >= 0 else -1 for x in x_values]

#Threshold is 0.6
def binary_step(x_values):
    return [1 if x >= 0.6 else 0 for x in x_values]

#Threshold is 0.6
def bipolar_step(x_values):
    return [1 if x >= 0.6 else -1 for x in x_values]

#Thresholds are 0.3 and 0.6
def saturating_linear_transfer(x_values):
    y_values = []
    for x in x_values:
        if x < 0.3: y_values.append(0)
        elif 0.3 <= x <= 0.6: y_values.append((x-0.3)/(0.6-0.3))
        elif x > 0.6: y_values.append(1)
    return y_values

def sigmoid_transfer(x_values):
    return [(1-exp(-2*x))/(1+exp(-2*x)) for x in x_values]

def log_sigmoid_transfer(x_values):
    return [1/(1+exp(-x)) for x in x_values]
```

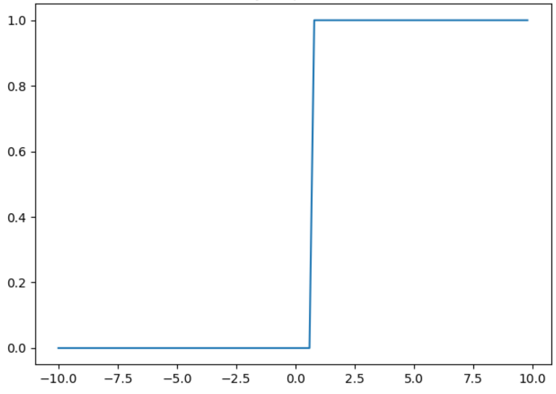
Plotting Graph of Activation Functions

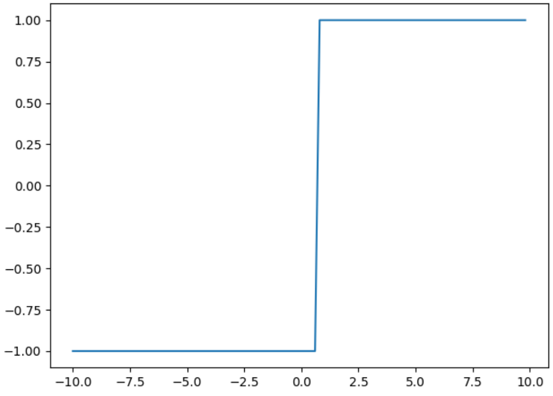
Same input is passed through all the different activation functions, and the function values are plotted

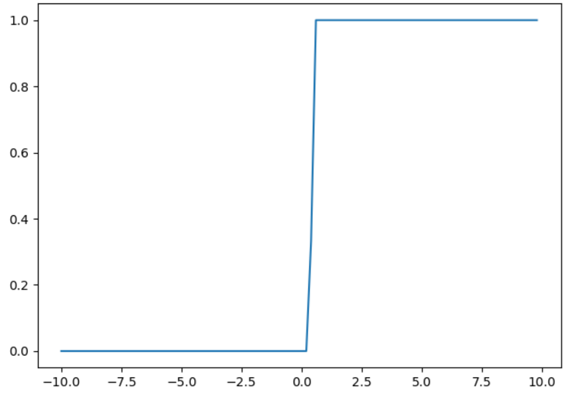
In [ ]:

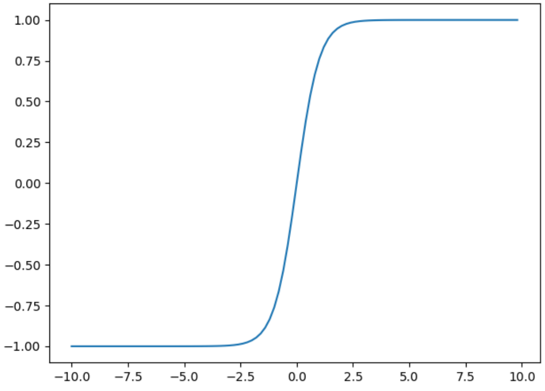
```
activation_functions = [
    'symmetric_hard_limit_transfer',
    'binary_step',
    'bipolar_step',
    'saturating_linear_transfer',
    'sigmoid_transfer',
    'log_sigmoid_transfer'
]
for activation_function in activation_functions:
    x_values = arange(-10., 10., 0.2)
    y_values = globals()[activation_function](x_values)
    title(' '.join([word.capitalize() for word in activation_function.spl
it('_')]+['Function']))
    plot(x_values, y_values)
    show()
```

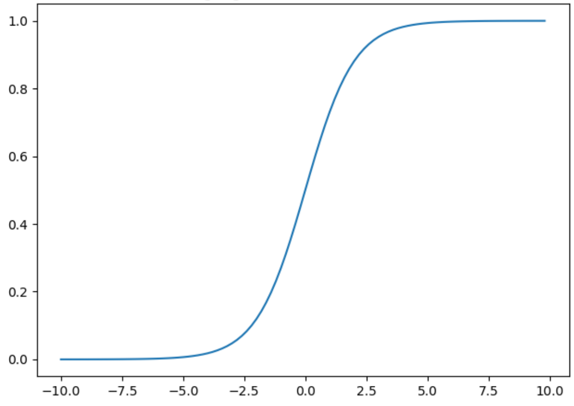

Symmetric Hard Limit Transfer Function


Binary Step Function


Bipolar Step Function


Saturating Linear Transfer Function


Sigmoid Transfer Function


Log Sigmoid Transfer Function