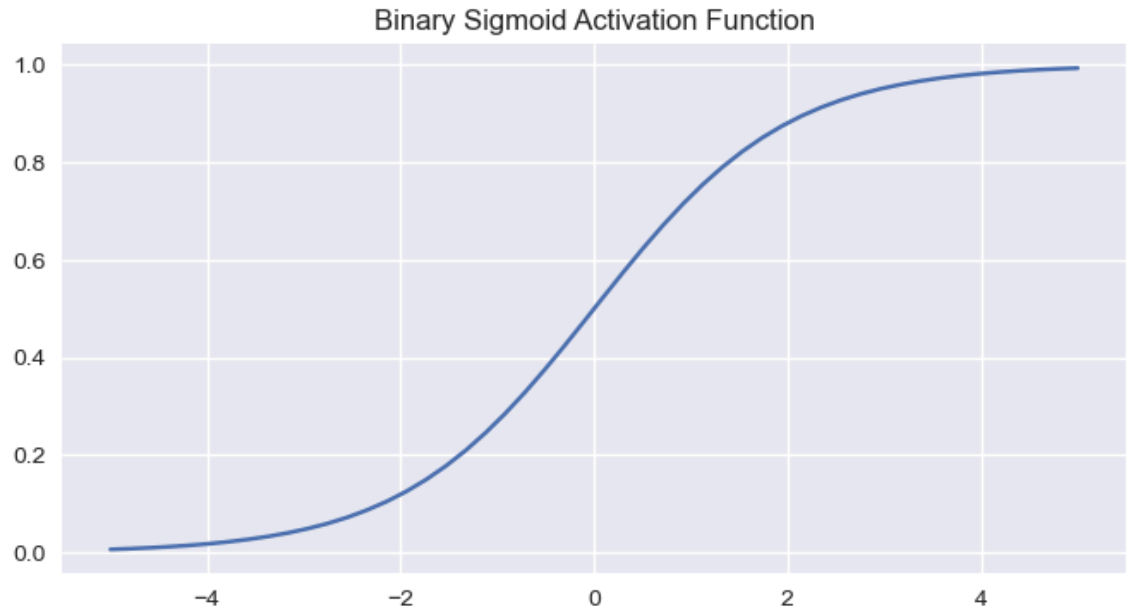
In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import numpy as np
plt.style.use('seaborn')
plt.figure(figsize=(8,4))
def SigmoidBinary(t):
    return 1/(1+np.exp(-t))
t = np.linspace(-5, 5)
plt.plot(t, SigmoidBinary(t))
plt.title('Binary Sigmoid Activation Function')
plt.show()
```

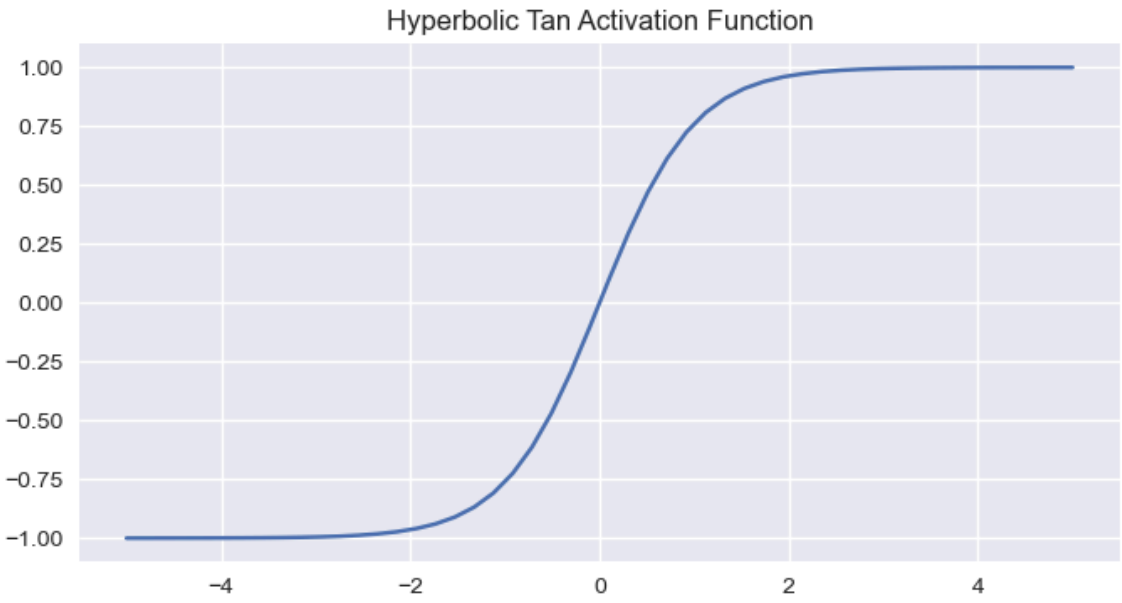
```
C:\Users\user\AppData\Local\Temp\ipykernel_15568\3850015010.py:4: MatplotlibDeprecationWarning: The seaborn styles shipped by Matplotlib are deprecated since 3.6, as they no longer correspond to the styles shipped by seaborn. However, they will remain available as 'seaborn-v0_8-<style>'. Alternatively, directly use the seaborn API instead.
  plt.style.use('seaborn')
```



In [2]:

```
import numpy as np
import matplotlib.pyplot as plt
import numpy as np
plt.style.use('seaborn')
plt.figure(figsize=(8,4))
def HyperbolicTan(t):
    return np.tanh(t)
t = np.linspace(-5, 5)
plt.plot(t, HyperbolicTan(t))
plt.title('Hyperbolic Tan Activation Function')
plt.show()
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_15568\1016814471.py:4: MatplotlibDeprecationWarning: The seaborn styles shipped by Matplotlib are deprecated since 3.6, as they no longer correspond to the styles shipped by seaborn. However, they will remain available as 'seaborn-v0_8-<style>'. Alternatively, directly use the seaborn API instead.
  plt.style.use('seaborn')
```

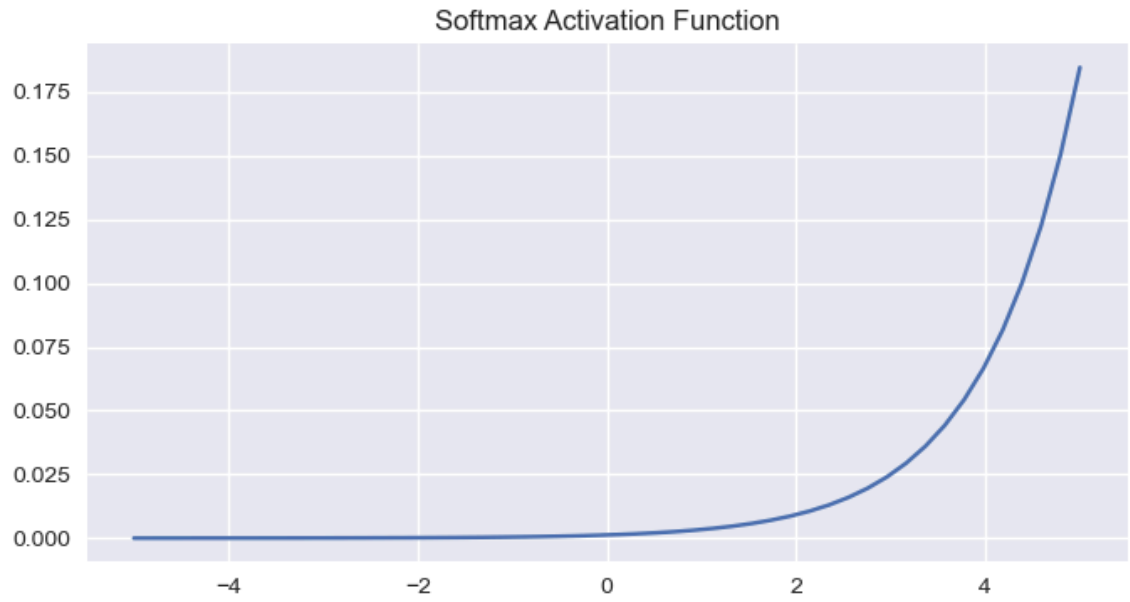


In [4]:

```
import numpy as np
import matplotlib.pyplot as plt
import numpy as np
plt.style.use('seaborn')
plt.figure(figsize=(8,4))
def softmax(t):
    return np.exp(t) / np.sum(np.exp(t))
t = np.linspace(-5, 5)
plt.plot(t, softmax(t))
plt.title('Softmax Activation Function')
plt.show()

```

```
C:\Users\user\AppData\Local\Temp\ipykernel_15568\2306039331.py:4: MatplotlibDeprecationWarning: The seaborn styles shipped by Matplotlib are deprecated since 3.6, as they no longer correspond to the styles shipped by seaborn. However, they will remain available as 'seaborn-v0_8-<style>'. Alternatively, directly use the seaborn API instead.
  plt.style.use('seaborn')
```

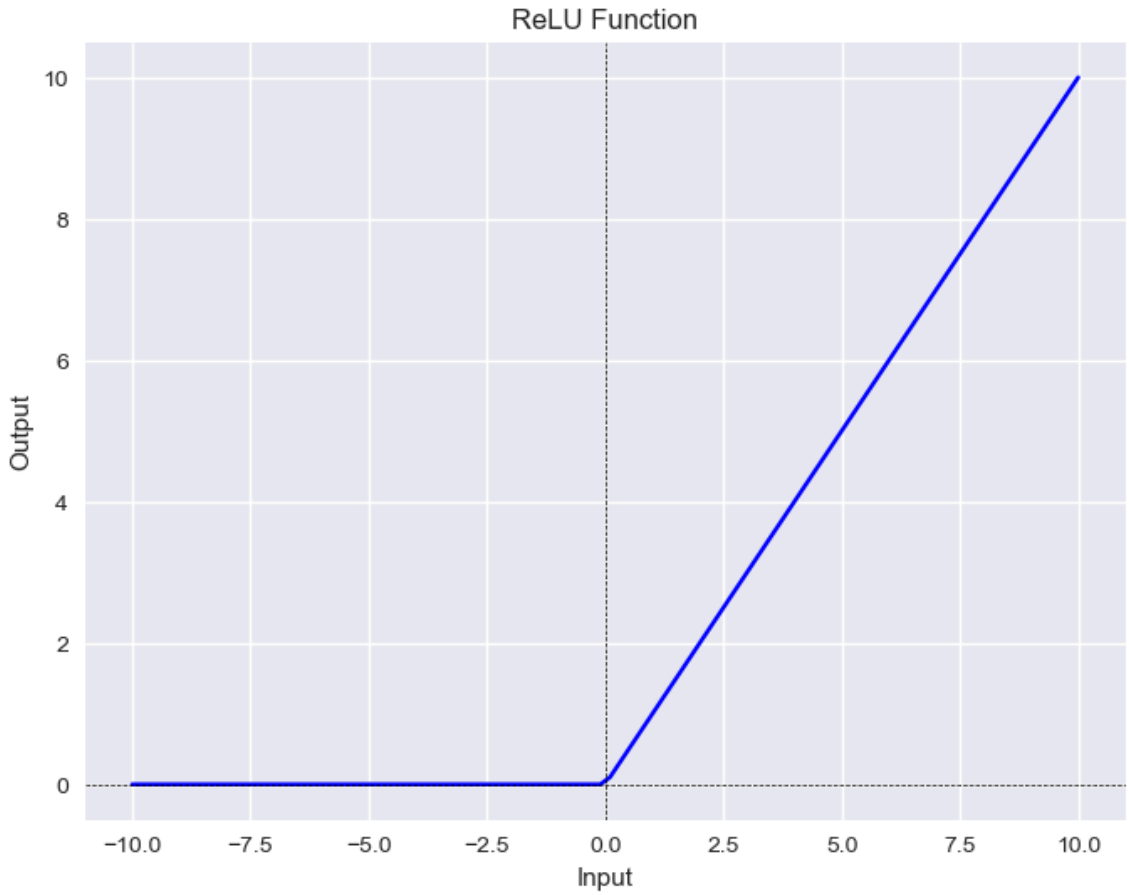


In [9]:

```
import numpy as np
import matplotlib.pyplot as plt
def relu(x):
    return np.maximum(0, x)
x = np.linspace(-10, 10, 100)
y = relu(x)
plt.figure(figsize=(8, 6))
plt.plot(x, y, color='blue')
plt.title('ReLU Function')
plt.xlabel('Input')
plt.ylabel('Output')
plt.axhline(0, color='black', linewidth=0.5, linestyle='--')  # Add x-a
plt.axvline(0, color='black', linewidth=0.5, linestyle='--')  # Add y-a
plt.legend()
plt.grid(True)
plt.show()

```

```
No artists with labels found to put in legend.  Note that artists whose la
bel start with an underscore are ignored when legend() is called with no a
rgument.
```



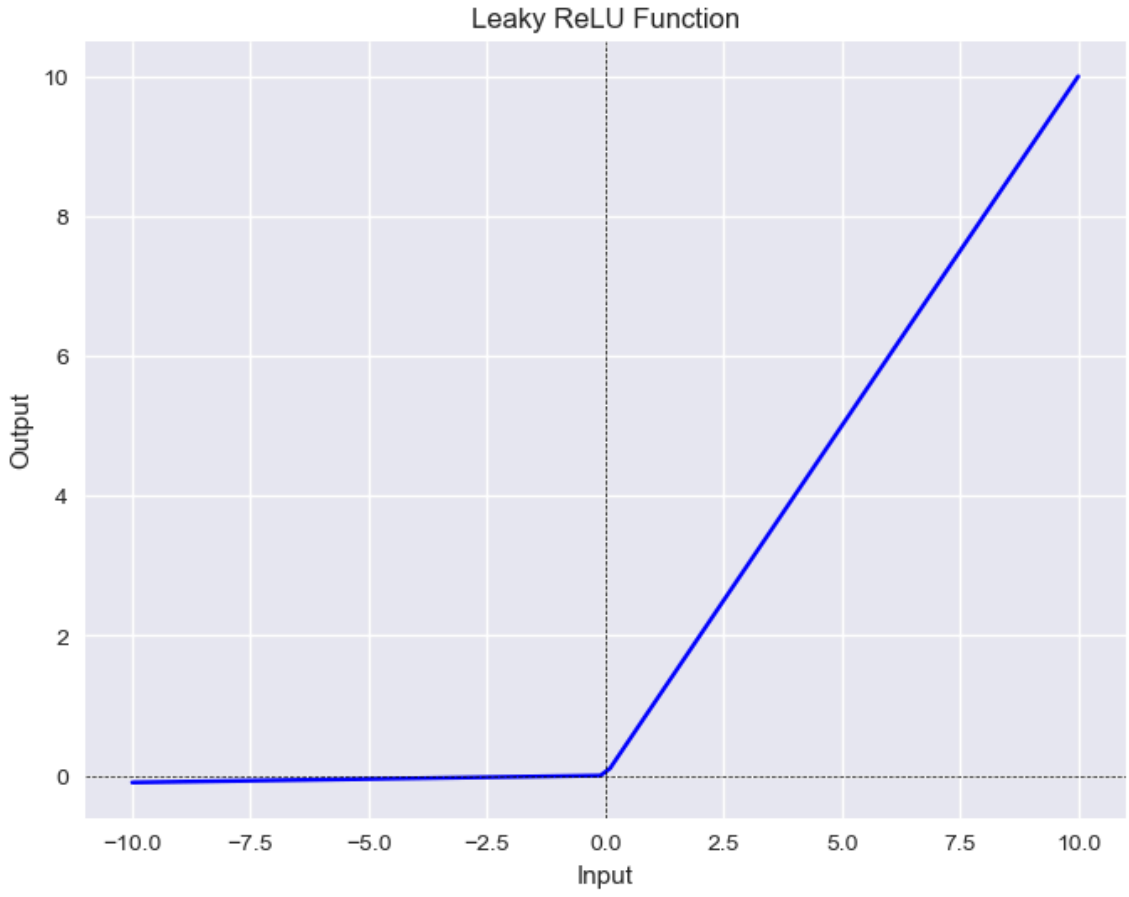
In [11]:

```python
import numpy as np
import matplotlib.pyplot as plt

def leaky_relu(x, alpha=0.01):
    return np.where(x > 0, x, alpha * x)
x = np.linspace(-10, 10, 100)
y = leaky_relu(x)
plt.figure(figsize=(8, 6))
plt.plot(x, y, color='blue')
plt.title('Leaky ReLU Function')
plt.xlabel('Input')
plt.ylabel('Output')
plt.axhline(0, color='black', linewidth=0.5, linestyle='--')  # Add x-a
plt.axvline(0, color='black', linewidth=0.5, linestyle='--')  # Add y-a
plt.legend()
plt.grid(True)
plt.show()

```

```
No artists with labels found to put in legend.  Note that artists whose la
bel start with an underscore are ignored when legend() is called with no a
rgument.
```



In [ ]:

```python

```