ACTIVATION FUNCTIONS

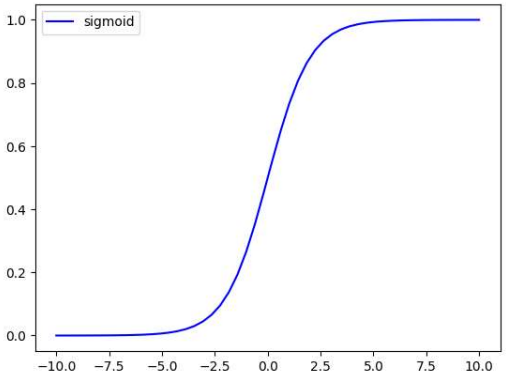
Sigmoid Function

```
f(x) = 1\(1+e^-x)
```

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

def sigmoid(x):
  return 1.0/(1.0+ np.exp(-x))

x = np.linspace(-10,10)
plt.plot(x,sigmoid(x), 'blue', label = 'sigmoid')
plt.legend()
plt.show()
```



Derivative of Sigmoid

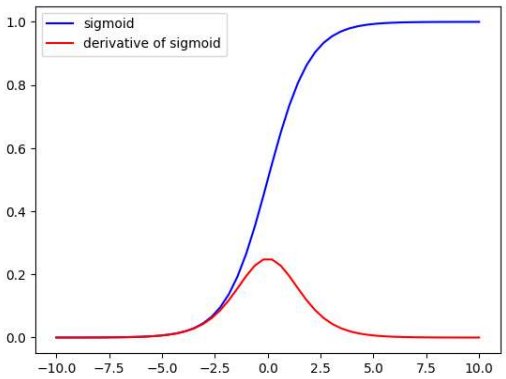
```
d(fx) = e^/(1+e^-x)^2
```

```
# plotting the derivative of sigmoid
import numpy as np
import matplotlib.pyplot as plt

def sigmoid(x):
  return 1.0/(1.0+ np.exp(-x))

def dSig(x):
  return np.exp(-x)/(1.0 + np.exp(-x))**2

x = np.linspace(-10,10)
plt.plot(x,sigmoid(x), 'blue', label = 'sigmoid')
plt.plot(x, dSig(x), 'red', label = 'derivative of sigmoid')
plt.legend()
plt.show()
```



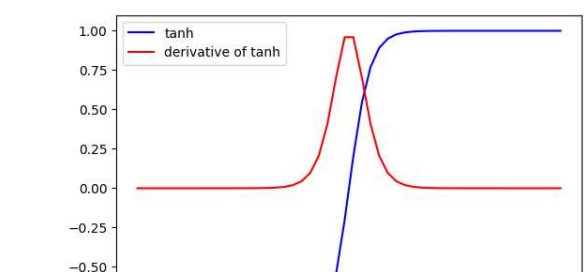
Tanh

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

def tanh(x):
  return (np.exp(x)-np.exp(-x))/(np.exp(x)+np.exp(-x))

def dTan(x):
  return (4*np.exp(-2*x)/(1.0 + np.exp(-2*x))**2)

x = np.linspace(-10,10)
plt.plot(x,tanh(x), 'blue', label = 'tanh')
plt.plot(x,dTan(x), 'red', label = 'derivative of tanh')
plt.legend()
plt.show()
```



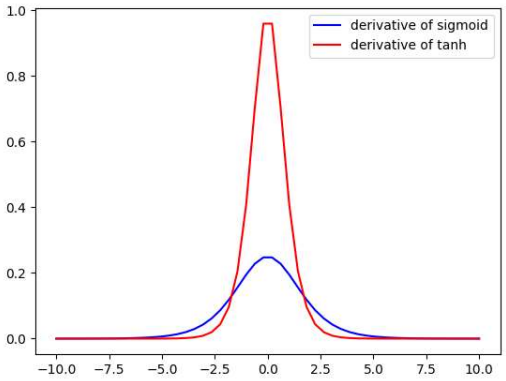
Derivatives of Sigmoid and Tanh

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

def dSig(x):
  return np.exp(-x)/(1.0 + np.exp(-x))**2

def dTan(x):
  return (4*np.exp(-2*x)/(1.0 + np.exp(-2*x))**2)

x = np.linspace(-10,10)
plt.plot(x,dSig(x), 'blue', label = 'derivative of sigmoid')
plt.plot(x,dTan(x), 'red', label = 'derivative of tanh')
plt.legend()
plt.show()
```



ReLu Function

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

def ReLu(x):
  d = map(lambda x: max(0,x),x)
  return np.array(list(d))
  # making a array from the list

x = np.linspace(-10,10)
plt.plot(x,ReLu(x), 'blue', label = 'ReLu')
plt.legend()
plt.show()
```