**Foundation of Data Engineering**

**Activation Functions:**

1. **Sigmoid function**

Code:  
import numpy as np

import matplotlib.pyplot as plt

# Define sigmoid function

def sigmoid(x):

    return 1 / (1 + np.exp(-x))

# Define derivative of sigmoid function

def sigmoid\_derivative(x):

    s = sigmoid(x)

    return s \* (1 - s)

# Generate input values

x = np.linspace(-10, 10, 1000)

# Compute sigmoid and its derivative

y\_sigmoid = sigmoid(x)

y\_derivative = sigmoid\_derivative(x)

# Plotting

plt.figure(figsize=(10, 5))

# Sigmoid Plot

plt.subplot(1, 2, 1)

plt.plot(x, y\_sigmoid, label="Sigmoid", color='green')

plt.title("Sigmoid Function")

plt.xlabel("x")

plt.ylabel("σ(x)")

plt.grid(True)

plt.legend()

# Derivative Plot

plt.subplot(1, 2, 2)

plt.plot(x, y\_derivative, label="Sigmoid Derivative", color='orange')

plt.title("Derivative of Sigmoid")

plt.xlabel("x")

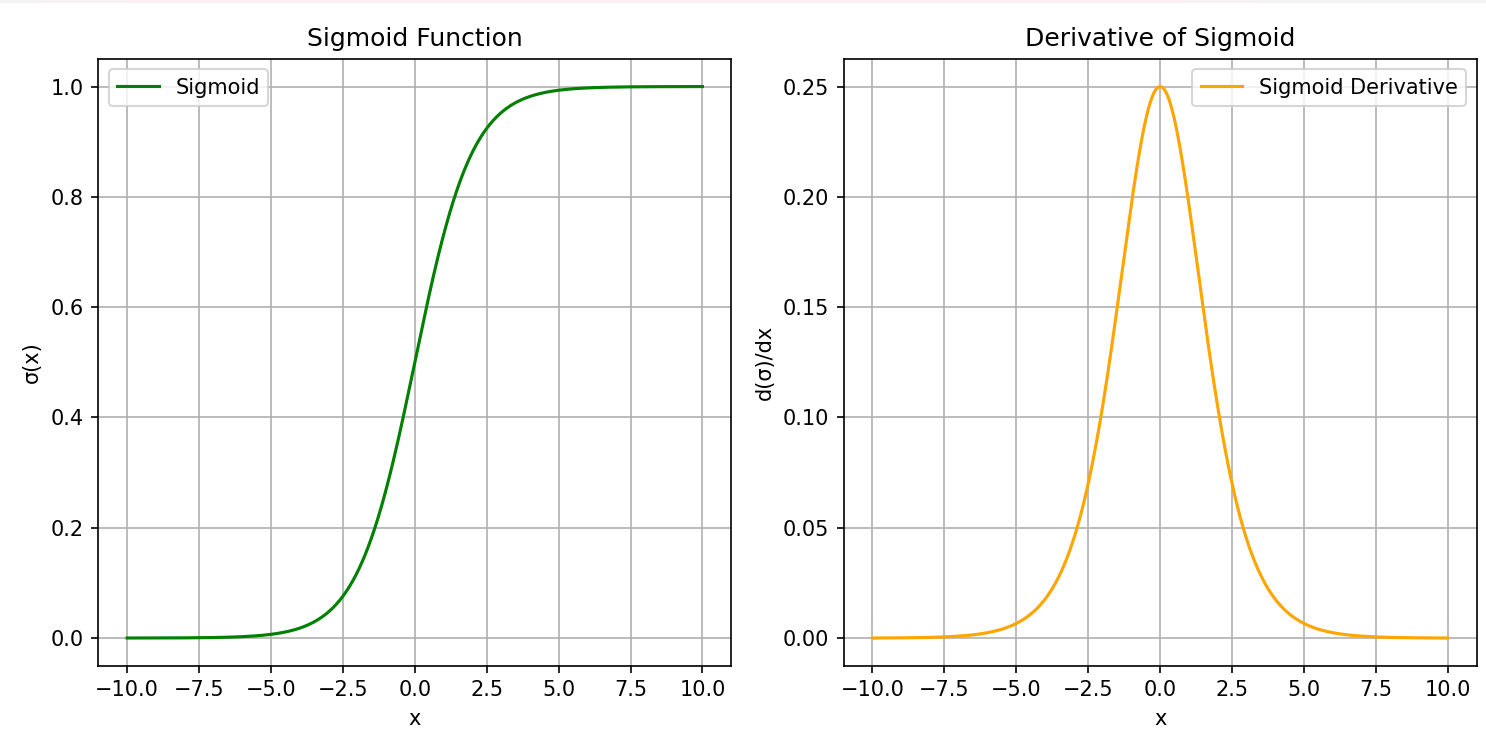
plt.ylabel("d(σ)/dx")

plt.grid(True)

plt.legend()

plt.tight\_layout()

plt.show()

Output:  


1. **Relu function:**

import numpy as np

import matplotlib.pyplot as plt

# Define ReLU function

def relu(x):

return np.maximum(0, x)

# Define derivative of ReLU

def relu\_derivative(x):

return np.where(x > 0, 1, 0)

# Generate input values

x = np.linspace(-10, 10, 1000)

# Compute ReLU and its derivative

y\_relu = relu(x)

y\_derivative = relu\_derivative(x)

# Plotting

plt.figure(figsize=(10, 5))

# ReLU Plot

plt.subplot(1, 2, 1)

plt.plot(x, y\_relu, label="ReLU", color='blue')

plt.title("ReLU Function")

plt.xlabel("x")

plt.ylabel("ReLU(x)")

plt.grid(True)

plt.legend()

# Derivative Plot

plt.subplot(1, 2, 2)

plt.plot(x, y\_derivative, label="ReLU Derivative", color='red')

plt.title("Derivative of ReLU")

plt.xlabel("x")

plt.ylabel("d(ReLU)/dx")

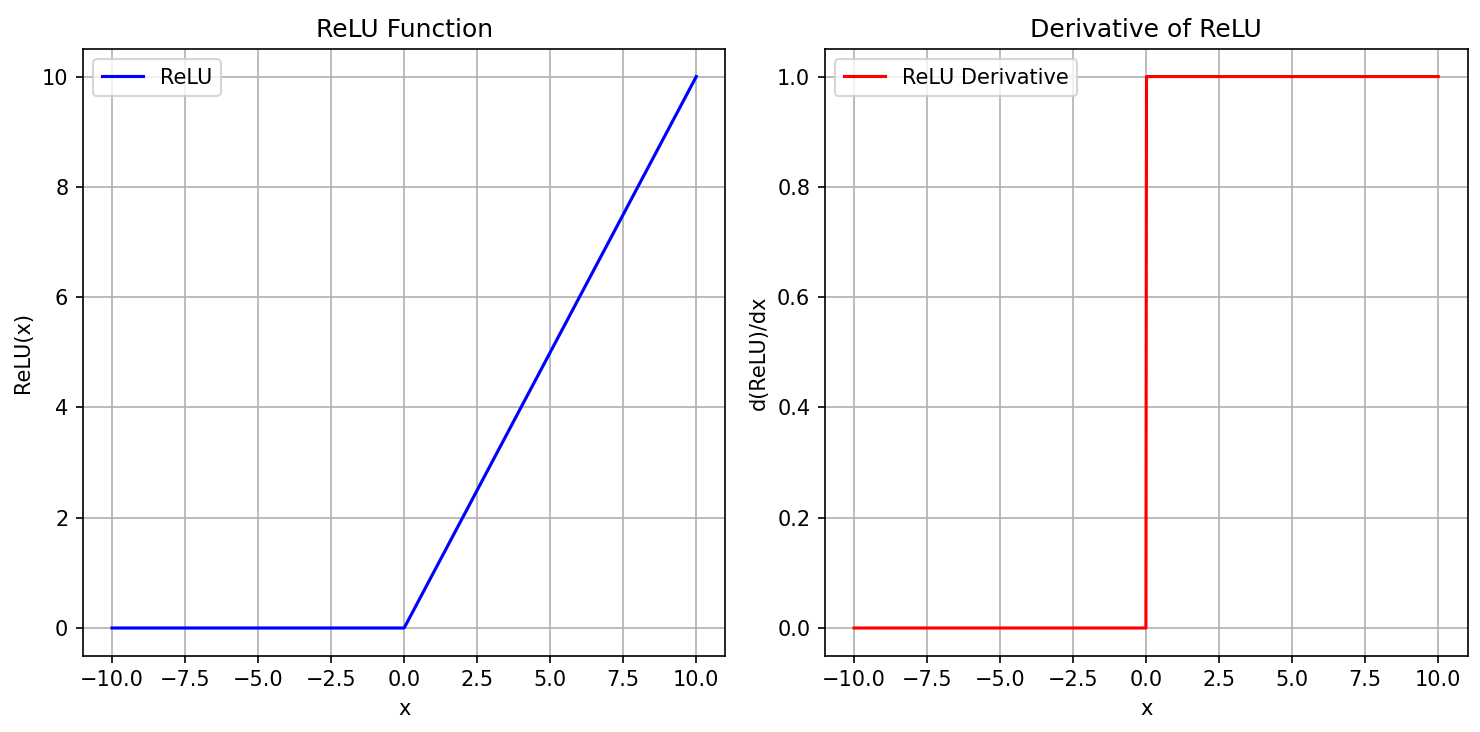
plt.grid(True)

plt.legend()

plt.tight\_layout()

plt.show()

Output:



1. **Tanh function:**

Code:

import numpy as np

import matplotlib.pyplot as plt

# Define tanh function

def tanh(x):

    return np.tanh(x)

# Define derivative of tanh function

def tanh\_derivative(x):

    return 1 - np.tanh(x)\*\*2

# Generate input values

x = np.linspace(-10, 10, 1000)

# Compute tanh and its derivative

y\_tanh = tanh(x)

y\_derivative = tanh\_derivative(x)

# Plotting

plt.figure(figsize=(10, 5))

# Tanh Plot

plt.subplot(1, 2, 1)

plt.plot(x, y\_tanh, label="Tanh", color='purple')

plt.title("Tanh Function")

plt.xlabel("x")

plt.ylabel("tanh(x)")

plt.grid(True)

plt.legend()

# Derivative Plot

plt.subplot(1, 2, 2)

plt.plot(x, y\_derivative, label="Tanh Derivative", color='brown')

plt.title("Derivative of Tanh")

plt.xlabel("x")

plt.ylabel("d(tanh)/dx")

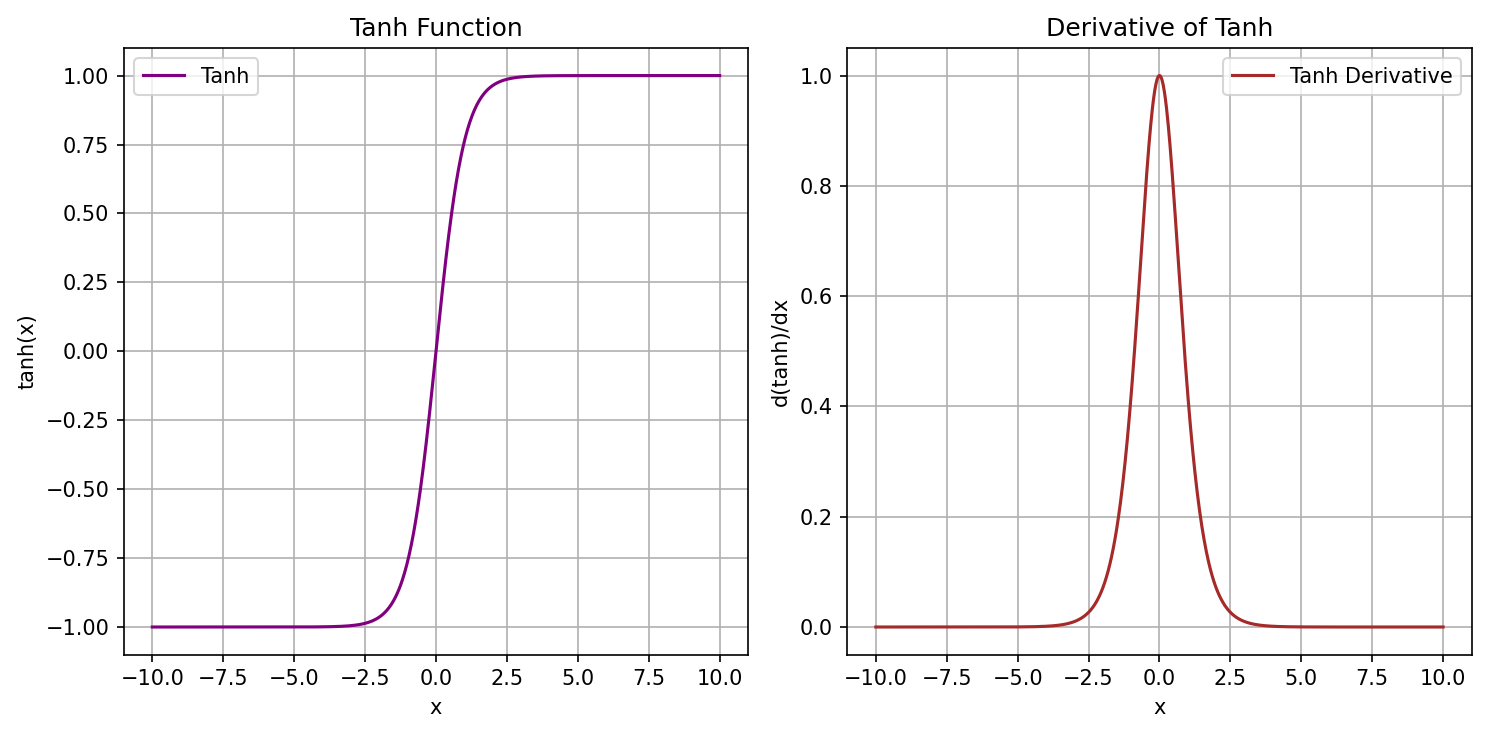
plt.grid(True)

plt.legend()

plt.tight\_layout()

plt.show()

**Output:**

****

**Basic Functions:  
1. f(x) = e^-x**

Code:

import numpy as np

import matplotlib.pyplot as plt

# Define the function f(x) = e^(-x)

def f(x):

    return np.exp(-x)

# Define its derivative f'(x) = -e^(-x)

def f\_derivative(x):

    return -np.exp(-x)

# Generate x values

x = np.linspace(-5, 5, 500)

# Compute y values

y = f(x)

y\_prime = f\_derivative(x)

# Plotting

plt.figure(figsize=(10, 5))

# f(x) Plot

plt.plot(x, y, label=r'$f(x) = e^{-x}$', color='blue')

# f'(x) Plot

plt.plot(x, y\_prime, label=r"$f'(x) = -e^{-x}$", color='red', linestyle='--')

# Formatting

plt.title("Function and Derivative: $f(x) = e^{-x}$")

plt.xlabel("x")

plt.ylabel("y")

plt.axhline(0, color='black', linewidth=0.5)

plt.axvline(0, color='black', linewidth=0.5)

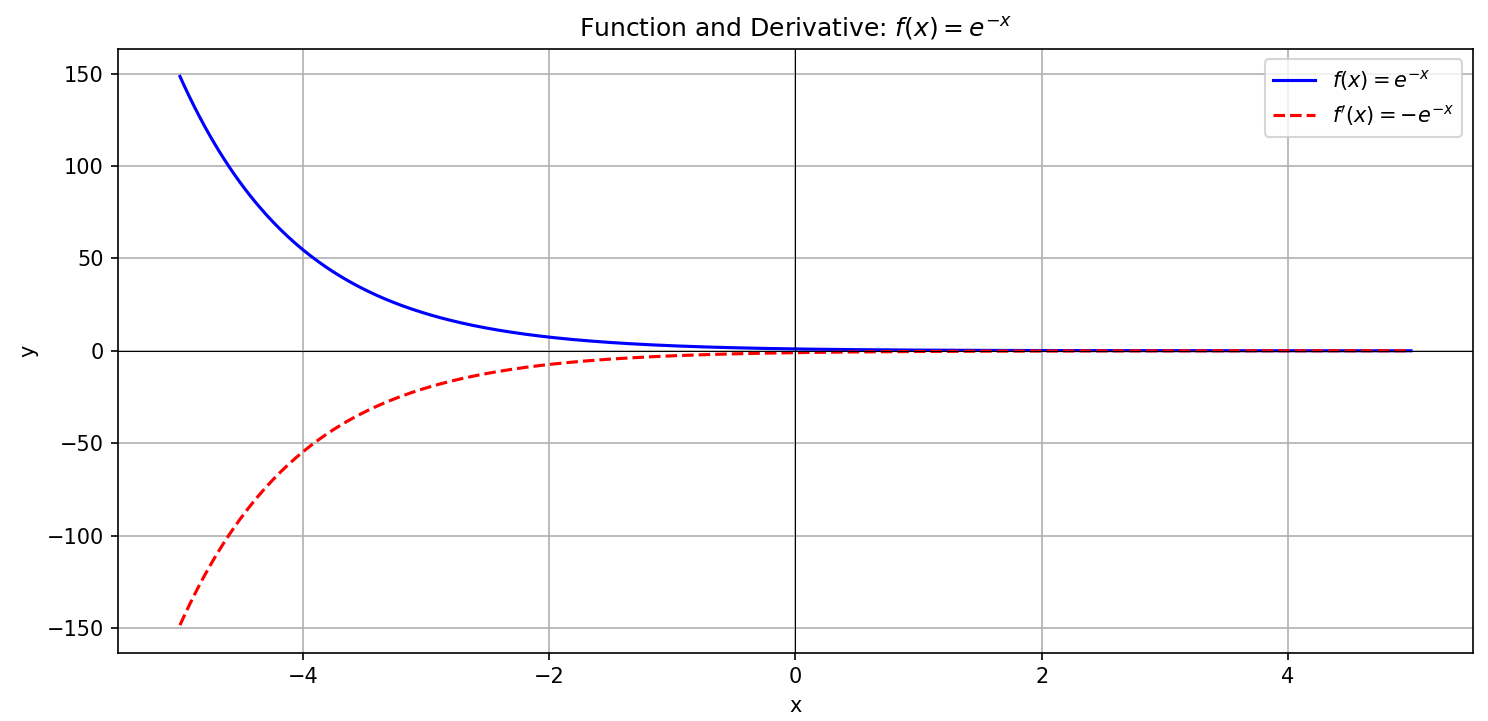
plt.grid(True)

plt.legend()

plt.tight\_layout()

plt.show()

Output:



**2. f(x) = e^-|x|**

Code:

import numpy as np

import matplotlib.pyplot as plt

# Define the function f(x) = e^(-|x|)

def f(x):

    return np.exp(-np.abs(x))

# Define the derivative of f(x)

def f\_derivative(x):

    return np.where(x > 0, -np.exp(-x), np.where(x < 0, np.exp(x), 0))  # 0 at x=0 for visualization

# Generate input values

x = np.linspace(-5, 5, 1000)

# Compute function and derivative

y = f(x)

y\_prime = f\_derivative(x)

# Plotting

plt.figure(figsize=(10, 5))

# Plot f(x)

plt.plot(x, y, label=r'$f(x) = e^{-|x|}$', color='blue')

# Plot f'(x)

plt.plot(x, y\_prime, label=r"$f'(x)$", color='orange', linestyle='--')

# Formatting

plt.title("Function and Derivative: $f(x) = e^{-|x|}$")

plt.xlabel("x")

plt.ylabel("y")

plt.grid(True)

plt.axhline(0, color='black', linewidth=0.5)

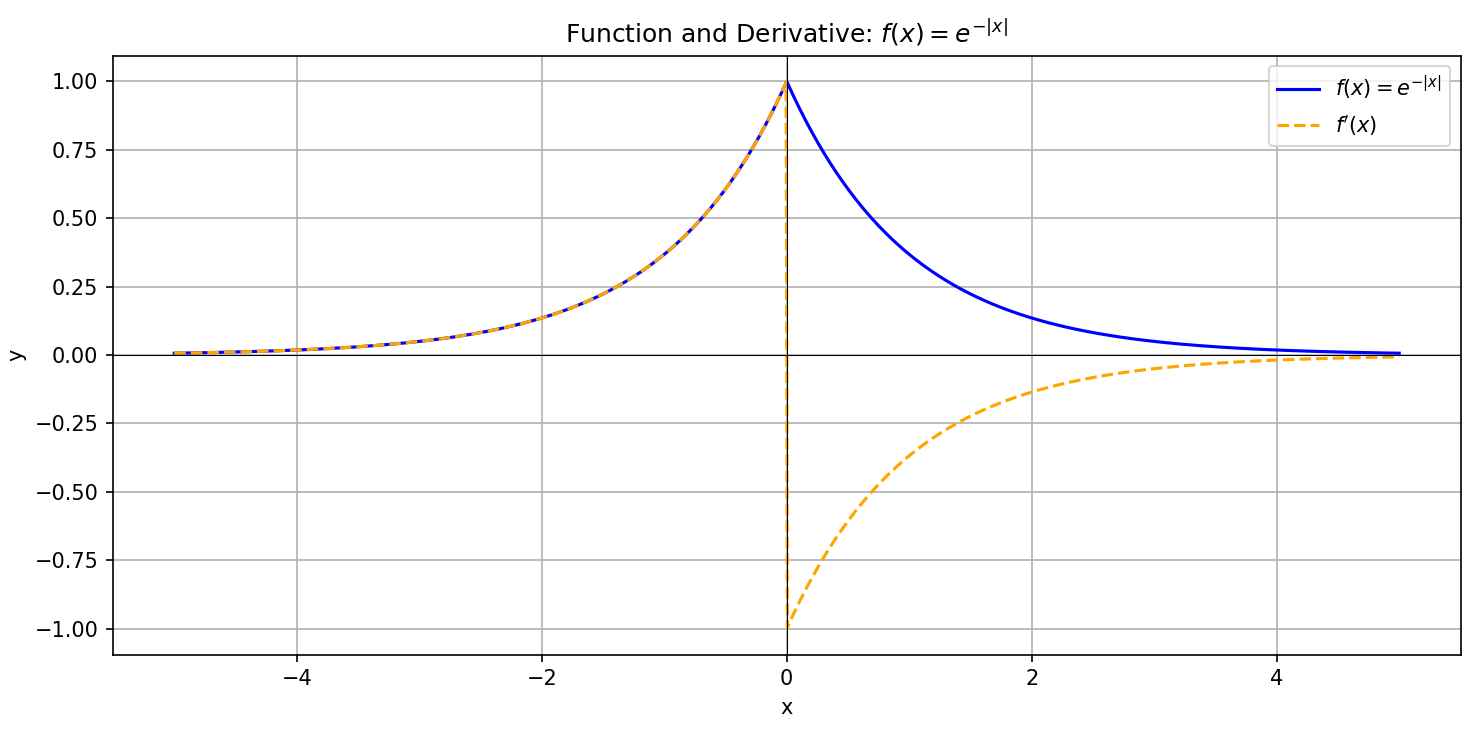
plt.axvline(0, color='black', linewidth=0.5)

plt.legend()

plt.tight\_layout()

plt.show()

Output:



**3.f(x) = e^-x^2**

Code:

import numpy as np

import matplotlib.pyplot as plt

# Define the function f(x) = e^(-x^2)

def f(x):

    return np.exp(-x\*\*2)

# Define its derivative f'(x) = -2x \* e^(-x^2)

def f\_derivative(x):

    return -2 \* x \* np.exp(-x\*\*2)

# Generate x values

x = np.linspace(-3, 3, 500)

# Compute y values

y = f(x)

y\_prime = f\_derivative(x)

# Plotting

plt.figure(figsize=(10, 5))

# Plot f(x)

plt.plot(x, y, label=r'$f(x) = e^{-x^2}$', color='blue')

# Plot f'(x)

plt.plot(x, y\_prime, label=r"$f'(x) = -2x e^{-x^2}$", color='red', linestyle='--')

# Formatting

plt.title("Function and Derivative: $f(x) = e^{-x^2}$")

plt.xlabel("x")

plt.ylabel("y")

plt.grid(True)

plt.axhline(0, color='black', linewidth=0.5)

plt.axvline(0, color='black', linewidth=0.5)

plt.legend()

plt.tight\_layout()

plt.show()

Output:

