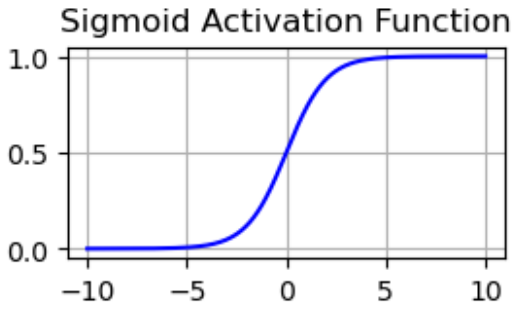
```
Aarya Admane
Roll no-22630
Div-B
Batch-B2
```

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

def sigmoid(x):
    return 1 / (1 + np.exp(-x))


def tanh(x):
    return np.tanh(x)

def relu(x):
    return np.maximum(0, x)


def leaky_relu(x, alpha=0.01):
    return np.where(x > 0, x, alpha * x)


def softmax(x):
    e_x = np.exp(x - np.max(x))
    return e_x / e_x.sum(axis=0, keepdims=True)


# Generate input data
x = np.linspace(-10, 10, 400)
x_reshaped = x.reshape(-1, 1)  # To make softmax work on it


# Plot the activation functions
plt.figure(figsize=(10, 8))
```

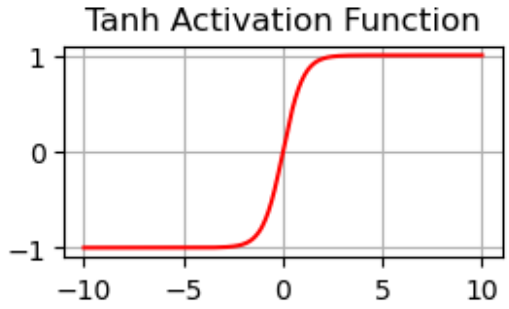
```
<Figure size 1000x800 with 0 Axes>
```

```
<Figure size 1000x800 with 0 Axes>
```

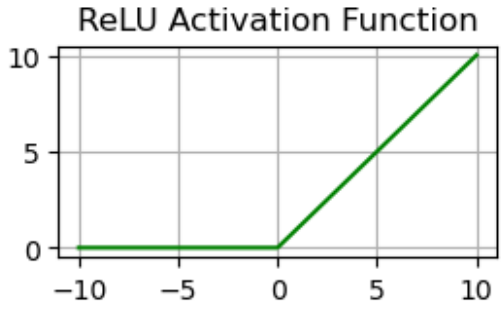
```
# Sigmoid
plt.subplot(3, 2, 1)
plt.plot(x, sigmoid(x), label='Sigmoid', color='b')
plt.title('Sigmoid Activation Function')
plt.grid(True)
```




```
# Tanh
plt.subplot(3, 2, 2)
plt.plot(x, tanh(x), label='Tanh', color='r')
plt.title('Tanh Activation Function')
plt.grid(True)
```

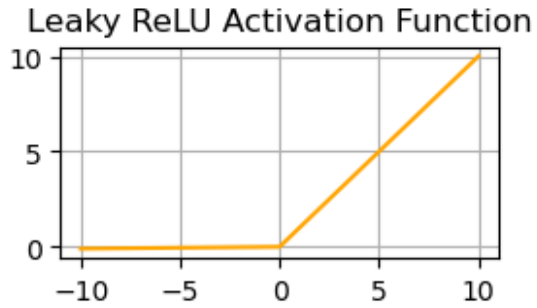



```
# ReLU
plt.subplot(3, 2, 3)
plt.plot(x, relu(x), label='ReLU', color='g')
plt.title('ReLU Activation Function')
plt.grid(True)
```

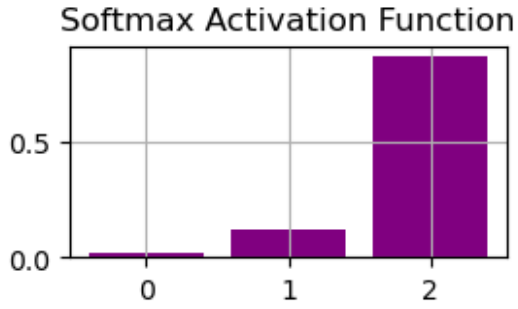



```
# Leaky ReLU
plt.subplot(3, 2, 4)
```

```
plt.plot(x, leaky_relu(x), label='Leaky ReLU', color='orange')
plt.title('Leaky ReLU Activation Function')
plt.grid(True)
```




```
# Softmax - Just a small demo for softmax with 3 inputs
x_softmax = np.linspace(-2, 2, 3).reshape(3, 1)
softmax_output = softmax(x_softmax)


plt.subplot(3, 2, 5)
plt.bar(np.arange(3), softmax_output.flatten(), color='purple')
plt.title('Softmax Activation Function')
plt.grid(True)
```




```
# Adjust layout for better spacing
plt.tight_layout()
plt.show()
```

```
<Figure size 640x480 with 0 Axes>
```