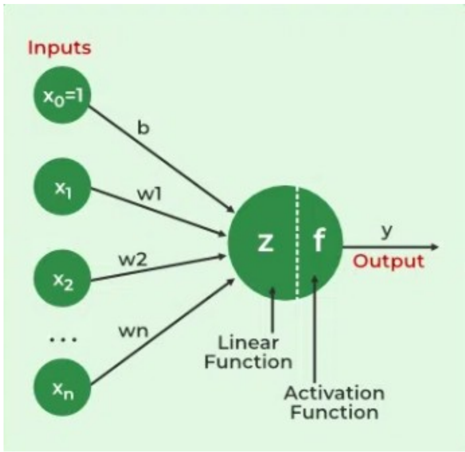
Feedforward Neural Network



```
# Lets Implement the basic Feed Forward Neural Network
import numpy as np

# Inputs (x1, x2, x3) and bias (b)
x1 = 1
x2 = 2
x3 = 3
b  = 0

# Weights
w1 = 0.21
w2 = 0.02

# implement (linear function Z )
# z = wx + b (formula)
z = w1 * x1 +  w2 * x2 + b


# Sigmoid Activation function
f = 1 / 1 + np.exp(-z)



print("Sigmoid activation function :", f)
```

```
Sigmoid activation function : 1.778800783071405
```

Implementation through Object Oriented Programming

```
class Feed_forward_Neural_network:
  def __init__(self, x1, x2, x3, w1, w2, b):  # Inputs and Weights
    self.x1 = x1
    self.x2 = x2
    self.x3 = x3
    self.w1 = w1
    self.w2 = w2
    self.b  = b

  def linear_function(self):                  # Linear Function z
    return self.w1 * self.x1 + self.w2 * self.x2 + self.b

  def sigmoid_activation_function(self):      # Activation Function
    z = self.linear_function()
    return np.clip(1 / 1 + np.exp(-z), 0, 1)

  def Output(self):                           # Output
    return self.sigmoid_activation_function()


obj = Feed_forward_Neural_network(0.1, 1, 2, 1.9, 0.4, 0)
print("Sigmoid activation function :", obj.Output())
```

```
Sigmoid activation function : 1.0
```