

# Activation Functions in Deep Learning

---

Common Activation Functions in ANNs are important in neural networks because they introduce non-linearity and helps the network to learn complex patterns. Lets see some common activation functions used in ANNs

---

1. **Sigmoid** Function: Outputs values between 0 and 1. It is used in binary classification tasks like deciding if an image is a cat or not.

---

2. **ReLU (Rectified Linear Unit)**: A popular choice for hidden layers, it returns the input if positive and zero otherwise. It helps to solve the vanishing gradient problem.

---

3. **Tanh (Hyperbolic Tangent)**: Similar to sigmoid but outputs values between -1 and 1. It is used in hidden layers when a broader range of outputs is needed.

---

4. **Softmax**: Converts raw outputs into probabilities used in the final layer of a network for multi-class classification tasks.

```
# Numpy for calculations
import numpy as np

# Sigmoid Function (The output from 0 to 1)
def sigmoid(x):
    sigma = 1 / (1 + np.exp(-x))
    return sigma

# ReLu f(x) = (0 , max)
def ReLu(x):
    relu = np.maximum(0, x)
    return relu

# Tanh the values between (-1 to +1)
def Tanh(x):
    tanh = np.sinh(x) / np.cosh(x)
    return tanh
```

```
# Softmax
def Softmax(z):
    softmax = np.exp(z) / np.exp(sum(z))
    return softmax

# Driver Activation function (All in one)
def Activation_functions(x):
    print("Sigmoid Activation--> ", sigmoid(x))
    print("ReLu Activation--> ", ReLu(x))
    print("Tanh Activation--> ", Tanh(x))
    print("Softmax Activation--> ", Softmax([1.2, 2.3, 3.2]))
```

```
Activation_functions(0.45)
```

```
Sigmoid Activation--> 1.6376281516217732
ReLu Activation--> 0.45
Tanh Activation--> 0.42189900525000795
Softmax Activation--> [0.00408677 0.01227734 0.03019738]
```