Assignment No: 8

Title: Program for creating back propogation feed-forward neural network.

Name : Tavhare Ruchita Sharad

Roll No:58

In [2]:
```
import numpy as np
```

In [4]:
```
def sigmoid(x):
    return 1/(1+np.exp(-x))
def sigmoid_derivative(x):
    return x*(1-x)
```

In [6]:
```
X =np.array([[0,0],[0,1],[1,0],[1,1]])
Y =np.array([[0],[1],[1],[0]])
```

In [8]:
```
learning_rate =0.1
num_epochs=100000
```

In [10]:
```
hidden_weights=2*np.random.random((2,2))-1
output_weights=2*np.random.random((2,1))-1
```

In [12]:
```
for i in range(num_epochs):
    hidden_layer = sigmoid(np.dot(X,hidden_weights))
    output_layer = sigmoid(np.dot(hidden_layer,output_weights))
    output_error = output_layer
    output_delta = output_error*sigmoid_derivative(output_layer)
    hidden_error = output_delta.dot(output_weights.T)
    hidden_delta = hidden_error*sigmoid_derivative(hidden_layer)
    output_weights+=hidden_layer.T.dot(output_delta)*learning_rate
    hidden_weights+=X.T.dot(hidden_delta)*learning_rate
```

In [13]:
```
print("Input:")
print(X)
print("output:")
print(output_layer)
```

```
Input:
[[0 0]
 [0 1]
 [1 0]
 [1 1]]
output:
[[0.99981067]
 [0.99999841]
 [0.99999889]
 [0.99999988]]
```

In [ ]:

In [ ]: