

EXPERIMENT – 4

Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.

CODE :

```
# ANN using Backpropagation (XOR)
```

```
import math
```

```
import random
```

```
# Sigmoid
```

```
def sigmoid(x):
```

```
    return 1 / (1 + math.exp(-x))
```

```
def d_sigmoid(y):
```

```
    return y * (1 - y)
```

```
# Training data
```

```
X = [[0,0],[0,1],[1,0],[1,1]]
```

```
Y = [0,1,1,0]
```

```
# Weights
```

```
random.seed(1)
```

```
w1, w2, w3, w4 = [random.random() for _ in range(4)]
```

```
w5, w6 = random.random(), random.random()
```

```
lr = 0.5
```

```
# Training
```

```
for _ in range(5000):
```

```
    for x, y in zip(X, Y):
```

```
        # Forward
```

```
        h1 = sigmoid(x[0]*w1 + x[1]*w2)
```

```

h2 = sigmoid(x[0]*w3 + x[1]*w4)
out = sigmoid(h1*w5 + h2*w6)

# Backprop
error = y - out
d_out = error * d_sigmoid(out)

w5 += lr * d_out * h1
w6 += lr * d_out * h2
d_h1 = d_out * w5 * d_sigmoid(h1)
d_h2 = d_out * w6 * d_sigmoid(h2)
w1 += lr * d_h1 * x[0]
w2 += lr * d_h1 * x[1]
w3 += lr * d_h2 * x[0]
w4 += lr * d_h2 * x[1]

# Testing
print("Testing:")
for x in X:
    h1 = sigmoid(x[0]*w1 + x[1]*w2)
    h2 = sigmoid(x[0]*w3 + x[1]*w4)
    out = sigmoid(h1*w5 + h2*w6)
    print(x, "=>", round(out, 2))

```

OUTPUT :

```

>>> === RESTART: C:\Users\lalit\AppData\Local\Programs\Python\Python313\exp-4.py ===
Testing:
[0, 0] => 0.11
[0, 1] => 0.83
[1, 0] => 0.82
[1, 1] => 0.18
>>> |

```