

**19BCE248**  
**Practical 8**  
**2CS501**

**AIM:-** AND gate using Perceptron Learning without using any inbuilt libraries.

**Dataset:** As such only we need to do for 2 input parameters so a maximum of 4 combinations can be made of 0 and 1 for and or gate implementation.

**Preprocessing:** As such no preprocessing is needed because we have only binary data and no textual data. Also, only 4 instances are present so no train test splitting is done.

**Process:** Here first of all no hidden layers are taken and neural consists of 2 inputs and 1 output. Forward passing is done by using  $y = w \cdot x + b$  and through output backward passing is done using

$b = b + (\text{rate} * (Y[i] - y_{\text{pred}}))$  and  $W = W + (\text{rate} * (Y[i] - y_{\text{pred}}) * (x.T))$  for some finite iteration. So NN learns through these and predicts with proper output.

```
for iteration in range(epchos):
    for i in range(X.shape[0]):
        x=X[i]
        y=np.dot(x,W.transpose())+b
        y_pred=0
        if y>0:
            y_pred=1
        b=b+(rate*(Y[i]-y_pred))
        W=W+(rate*(Y[i]-y_pred)*(x.T))
    print("W ",W," Bias ",b)
```

Forward + Back Propagation.

**Conclusion:**

Through this experiment, we learned about how to build a NN from using without using any inbuilt libraries. And also a scratch implementation of NN enhanced our learning and made concepts more clear with respect to what an inbuilt library does.

