

```
In [1]: import numpy as np
        from random import randint
```

```
In [2]: X = np.array([[1,1,1],[1,-1,1],[-1,1,1],[-1,-1,1]])
        #output
        Y_ad=np.array([[1],[-1],[-1],[-1]])
        Y_o=np.array([[1],[1],[1],[-1]])
```

```
In [17]: print('input is:')
        print(X)
        print('output for And Gate is: ')
        print(Y_ad)
        print('output for Or Gate is: ')
        print(Y_o)
        weights_ad=np.zeros((3))
        weights_o=np.zeros((3))
        print(weights_ad)
        print(weights_o)
```

```
input is:
[[ 1  1  1]
 [ 1 -1  1]
 [-1  1  1]
 [-1 -1  1]]
output for And Gate is:
[[ 1]
 [-1]
 [-1]
 [-1]]
output for Or Gate is:
[[ 1]
 [-1]
 [-1]
 [-1]]
[0. 0. 0.]
[0. 0. 0.]
```

```
In [18]: # update weight for and gate /Logic
        def update_weight_ad(X,Y,weights):
            for i in range(4):

                weights=weights+X[i]*Y[i]

            #print weights
            slope =-(weights[0]/weights[1])
            c=-(weights[2]/weights[0])
            if slope<0 and weights[0]>0:
                weights_main=weights

            return weights_main
```

```
In [19]: def update_weight_o(X,Y,weights):
          for i in range(4):

              weights=weights+X[i]*Y[i]

              #print weights
              slope =-(weights[0]/weights[1])
              c=-(weights[2]/weights[0])
              if slope<0 and weights[0]>0:
                  weights_main=weights

          return weights_main
```

```
In [20]: weights_ad=update_weight_ad(X,Y_ad,weights_ad)
          weights_o=update_weight_o(X,Y_o,weights_o)
```

C:\Users\aj240\AppData\Local\Temp\ipykernel_3824\4010200372.py:9: RuntimeWarning: invalid value encountered in double_scalars
 c=-(weights[2]/weights[0])
 C:\Users\aj240\AppData\Local\Temp\ipykernel_3824\2802775205.py:7: RuntimeWarning: divide by zero encountered in double_scalars
 slope =-(weights[0]/weights[1])

```
In [21]: print('Checking after learning selectg a input')
          rand_int = int(input('Enter the test case no you want to try'))
          print('Select a logic you also want to check')
          logicgate = input()
          print(weights_ad)
          print('selected input is %d'%rand_int)
          print(X[rand_int])
```

Checking after learning selectg a input
 Enter the test case no you want to try1
 Select a logic you also want to check
 2
 [2. 2. -2.]
 selected input is 1
 [1 -1 1]

```
In [22]: def check_learning(X,weights,rand_int):
          Yin=0
          for i in range(3):
              print(X[rand_int,i])
              print(weights[i])
              Yin += X[rand_int,i]*weights[i]
          if Yin<0:
              Yin=-1
          else:
              Yin=1
          return Yin
```

```
In [23]: if logicgate=='a' or logicgate=='A':  
        weights_in = weights_ad  
        else:  
            weights_in = weights_o  
        Yin = check_learning(X, weights_in, rand_int)  
        print(Yin)
```

```
1  
2.0  
-1  
2.0  
1  
2.0  
1
```

```
In [24]: print('implemented by Aarya Jha on 15/11/22')
```

```
implemented by Aarya Jha on 15/11/22
```

```
In [ ]:
```