In [176]:
```python
import pandas as pd
import numpy as np
```

In [177]:
```python
x=[-1,-1,1,1]
y=[-1,1,-1,1]
z=[-1,-1,-1,1]
w=np.zeros(4)
theta=2
local_error=0.0
global_error=0.0
learning_rate=.1

def activation(m):
    if m>=theta:
        return 1
    else:
        return -1
```

In [178]:
```python
def cal(a,b):
    summ= a*w[0]+b*w[1]+w[2]
    return activation(summ)
```

In [179]:
```python
for i in range(100):
    global_error=0.0
    for j in range(4):
        output=cal(x[j],y[j])
        localError=z[j]-output
        w[0]+=learning_rate*localError*x[j]
        w[1]+=learning_rate*localError*y[j]
        w[2]+=learning_rate*localError
        global_error+=localError*localError
    if global_error==0:
        break
```

In [180]:
```python
for i in range(4):
    print(cal(x[i],y[i]))
```

```
-1
-1
-1
1
```

In [ ]:

In [ ]: