In [2]:
```
import numpy as np
import pandas as pd
from sklearn.metrics import accuracy_score
```

In [5]:
```
data = np.array([[0, 0, 0],
                 [0, 1, 1],
                 [1, 0, 1],
                 [1, 1, 1]])
```

In [6]:
```
X = data[:, :-1]
Y = data[:, -1]
print(X)
print()
print(Y)
```

```
[[0 0]
 [0 1]
 [1 0]
 [1 1]]

[0 1 1 1]
```

In [13]:
```
X_test = np.array([[0, 0],
                   [1, 1],
                   [1, 0]])

Y_test = np.array([0, 1, 1])
```

In [14]:
```
class Perceptron:
    def __init__ (self):
        self.w = None
        self.b = None

    def model(self, x):
        return 1 if (np.dot(self.w, x) >= self.b) else 0

    def predict(self, X):
        Y = []
        for x in X:
            result = self.model(x)
            Y.append(result)
        return np.array(Y)

    def fit(self, X, Y, epochs = 1, lr = 1):
        self.w = np.ones(X.shape[1])
        self.b = 0

        accuracy = {}
        max_accuracy = 0
        wt_matrix = []

        for i in range(epochs):
            for x, y in zip(X, Y):
                y_pred = self.model(x)
                if y == 1 and y_pred == 0:
                    self.w = self.w + lr * x
                    self.b = self.b - lr * 1
                elif y == 0 and y_pred == 1:
                    self.w = self.w - lr * x
                    self.b = self.b + lr * 1

            wt_matrix.append(self.w)

            accuracy[i] = accuracy_score(self.predict(X), Y)
            if (accuracy[i] > max_accuracy):
                max_accuracy = accuracy[i]
                chkptw = self.w
                chkptb = self.b

        self.w = chkptw
        self.b = chkptb

        print(max_accuracy)
        return np.array(wt_matrix)
```

In [15]:
```
perceptron = Perceptron()
```

In [16]:
```
perceptron.fit(X, Y)
```

```
1.0
```

Out[16]:
```
array([[1., 1.]])
```

In [18]:
```
result = perceptron.predict(X_test)
print(result)
```

```
[0 1 1]
```

In [19]:
```
print('Accuracy: ', accuracy_score(result, Y_test))
```

```
Accuracy:  1.0
```