In [1]:
```
import numpy as np
```

In [2]:
```
class Perceptron(object):
  def __init__(self, input_size, lr=0.2, epochs=4):
     self.W = np.array([0.3,-0.2])
     self.epochs = epochs
     self.lr = lr
```

In [3]:
```
  def activation_fn(self, x):
    return 1 if x >= 0 else 0
```

In [4]:
```
  def predict(self, x, theta):
    z = self.W.T.dot(x)-theta
    z=round(z,2)
    a = self.activation_fn(z)
    return a
```

In [5]:
```
  def fit(self, X, d,theta ,count):
    for _ in range(self.epochs):
      print("Epoch: ", count, "\n")
      count = count+1
      for i in range(d.shape[0]):
        x = X[i]
        print("input", x , "\t", "Weight:",self.W )
        print("\n")
        y = self.predict(x,theta)
        e = d[i] - y
        self.W = self.W + self.lr * e * x
```

In [6]:
```
if __name__ == '__main__':
   X = np.array([
     [0, 0],
     [0, 1],
     [1, 0],
     [1, 1]
   ])
   d = np.array([0, 1, 1, 1])
```

In [7]:
```
perceptron = Perceptron(input_size=2)
theta=0.4
count =1
perceptron.fit(X, d,theta, count)
print(perceptron.W)
```

```
---------------------------------------------------------------------
--------
AttributeError                            Traceback (most recent ca
ll last)
<ipython-input-7-582edc9767bb> in <module>
      2 theta=0.4
      3 count =1
----> 4 perceptron.fit(X, d,theta, count)
      5 print(perceptron.W)

AttributeError: 'Perceptron' object has no attribute 'fit'
```

In [12]:
```
import numpy as np
class Perceptron(object):
   def __init__(self, input_size, lr=0.2, epochs=4):
     self.W = np.array([0.3,-0.2])
     self.epochs = epochs
     self.lr = lr
   def activation_fn(self, x):
     return 1 if x >= 0 else 0
   def predict(self, x,theta):
     z = self.W.T.dot(x)-theta
     z=round(z,2)
     a = self.activation_fn(z)
     return a
   def fit(self, X, d,theta ,count):
     for _ in range(self.epochs):
       print("Epoch: ", count)
       count = count+1
       for i in range(d.shape[0]):
         x = X[i]
         print("input", x , "\t", "Weight:",self.W )
         y = self.predict(x,theta)
         e = d[i] - y
         self.W = self.W + self.lr * e * x
if __name__ == '__main__':
    X = np.array([
    [0, 0],
    [0, 1],
    [1, 0],
    [1, 1]
    ])
    d = np.array([0, 0, 0, 1])
    z = np.array([0, 1, 1, 1])
    y = np.array([0, 1, 1, 0])
    perceptron = Perceptron(input_size=2)
    theta=0.4
    count =1
    perceptron.fit(X, d,theta, count)
    print(perceptron.W)
```

```
Epoch:  1
input [0 0]      Weight: [ 0.3 -0.2]
input [0 1]      Weight: [ 0.3 -0.2]
input [1 0]      Weight: [ 0.3 -0.2]
input [1 1]      Weight: [ 0.3 -0.2]
Epoch:  2
input [0 0]      Weight: [0.5 0. ]
input [0 1]      Weight: [0.5 0. ]
input [1 0]      Weight: [0.5 0. ]
input [1 1]      Weight: [0.3 0. ]
Epoch:  3
input [0 0]      Weight: [0.5 0.2]
input [0 1]      Weight: [0.5 0.2]
input [1 0]      Weight: [0.5 0.2]
input [1 1]      Weight: [0.3 0.2]
Epoch:  4
input [0 0]      Weight: [0.3 0.2]
input [0 1]      Weight: [0.3 0.2]
input [1 0]      Weight: [0.3 0.2]
input [1 1]      Weight: [0.3 0.2]
```

[0 3 0 2]

In [13]:
```
perceptron.fit(X, z,theta, count)
print(perceptron.W)
```

```
Epoch:  1
input [0 0]      Weight: [0.3 0.2]
input [0 1]      Weight: [0.3 0.2]
input [1 0]      Weight: [0.3 0.4]
input [1 1]      Weight: [0.5 0.4]
Epoch:  2
input [0 0]      Weight: [0.5 0.4]
input [0 1]      Weight: [0.5 0.4]
input [1 0]      Weight: [0.5 0.4]
input [1 1]      Weight: [0.5 0.4]
Epoch:  3
input [0 0]      Weight: [0.5 0.4]
input [0 1]      Weight: [0.5 0.4]
input [1 0]      Weight: [0.5 0.4]
input [1 1]      Weight: [0.5 0.4]
Epoch:  4
input [0 0]      Weight: [0.5 0.4]
input [0 1]      Weight: [0.5 0.4]
input [1 0]      Weight: [0.5 0.4]
input [1 1]      Weight: [0.5 0.4]
[0.5 0.4]
```

In [15]:
```
perceptron.fit(X, y,theta, count)
print(perceptron.W)
```

```
Epoch:  1
input [0 0]      Weight: [0.5 0.4]
input [0 1]      Weight: [0.5 0.4]
input [1 0]      Weight: [0.5 0.4]
input [1 1]      Weight: [0.5 0.4]
Epoch:  2
input [0 0]      Weight: [0.3 0.2]
input [0 1]      Weight: [0.3 0.2]
input [1 0]      Weight: [0.3 0.4]
input [1 1]      Weight: [0.5 0.4]
Epoch:  3
input [0 0]      Weight: [0.3 0.2]
input [0 1]      Weight: [0.3 0.2]
input [1 0]      Weight: [0.3 0.4]
input [1 1]      Weight: [0.5 0.4]
Epoch:  4
input [0 0]      Weight: [0.3 0.2]
input [0 1]      Weight: [0.3 0.2]
input [1 0]      Weight: [0.3 0.4]
input [1 1]      Weight: [0.5 0.4]
[0.3 0.2]
```

In [ ]: