```
# -*- coding: utf-8 -*-
"""
Created on Thu Mar 23 12:29:55 2023

@author: tiver
"""

import numpy as np

class Perceptron:
    def __init__(self, input_size, lr=1, epochs=100):
        self.W = np.zeros(input_size+1)
        self.epochs = epochs
        self.lr = lr

    def activation_fn(self, x):
        return 1 if x >= 0 else 0

    def predict(self, x):
        z = self.W.T.dot(x)
        a = self.activation_fn(z)
        return a

    def fit(self, X, d):
        for epoch in range(self.epochs):
            for i in range(d.shape[0]):
                x = np.insert(X[i], 2, 1)
                y = self.predict(x)
                e = d[i] - y
                self.W = self.W + self.lr * e * x

#For AND Gate
X1 = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
d1 = np.array([0, 0, 0, 1])

#For OR Gate
X2 = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
d2 = np.array([0, 1, 1, 1])

perceptron1 = Perceptron(input_size=2)
perceptron1.fit(X1, d1)


perceptron2 = Perceptron(input_size=2)
perceptron2.fit(X2, d2)

print(perceptron1.W)
# Output: [ 2.  1. -3.]
print(perceptron2.W)
# Output: [ 1.  1. -1.]

test_in=np.array([0, 1, 1])
AND_prediction=perceptron1.predict(test_in)
print(AND_prediction)

OR_prediction=perceptron2.predict(test_in)
print(OR_prediction)
```

```
[ 2.  1. -3.]
[ 1.  1. -1.]
0
1
```