

```

import numpy as np

class Perceptron(object):

    def __init__(self, no_of_inputs, threshold=100, learning_rate=0.01):
        self.threshold = threshold
        self.learning_rate = learning_rate
        self.weights = np.zeros(no_of_inputs + 1)

    def predict(self, inputs):
        summation = np.dot(inputs, self.weights[1:]) + self.weights[0]
        if summation > 0:
            activation = 1
        else:
            activation = 0
        return activation

    def train(self, training_inputs, labels):
        for _ in range(self.threshold):
            for inputs, label in zip(training_inputs, labels):
                prediction = self.predict(inputs)
                self.weights[1:] += self.learning_rate * (label - prediction) * inputs
                self.weights[0] += self.learning_rate * (label - prediction)

def main():
    training_inputs= np.array([[1,2,3],[4,5,6],[7,8,9]])

    labels=np.array([0,1,1])
    perceptron=Perceptron(no_of_inputs=3)
    perceptron.train(training_inputs, labels)

    print("REG NO : 20MAI0018")
    inputs = np.array([1,2,3])
    print(perceptron.predict(inputs))

    inputs = np.array([4,5,6])
    print(perceptron.predict(inputs))

    inputs = np.array([4,7,8])
    print(perceptron.predict(inputs))

main()

```

```

☐→ REG NO : 20MAI0018
    0
    1
    0

```

