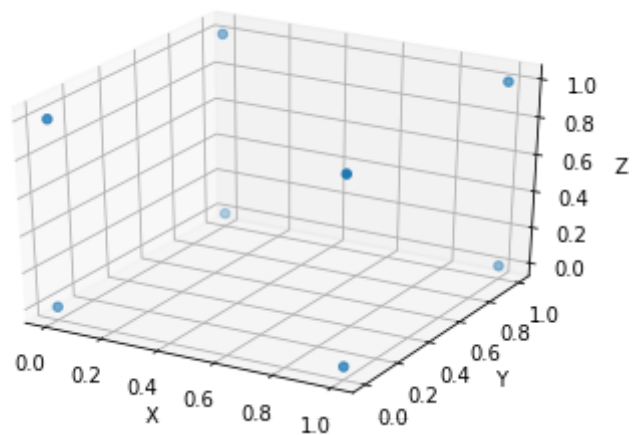


模式分类作业9_23

一. 题目描述

- 编写感知器算法程序，求下列模式分类的解向量：
- $w_1: \{(0\ 0\ 0), (1\ 0\ 0), (1\ 0\ 1), (1\ 1\ 0)\}$
- $w_2: \{(0\ 0\ 1), (0\ 1\ 1), (0\ 1\ 0), (1\ 1\ 1)\}$
- 设 $w(1)=(-1\ -2\ -2\ 0)$

二. 数据分析



将八个散点在坐标轴上的位置绘制出来

三. 程序运行结果

```
C:\Users\user\AppData\Local\Programs\Python\Python36\python.exe E:/MachineLearning/perceptron/perceptron.py
```

```
误分类点  w  b
[array([0, 0, 0]), array([-1, -2, -2]), 1]
[array([1, 0, 0]), array([ 0, -2, -2]), 2]
[array([1, 0, 1]), array([ 1, -2, -1]), 3]
[array([0, 0, 1]), array([ 1, -2, -2]), 2]
[array([0, 0, 1]), array([ 1, -2, -3]), 1]
[array([1, 0, 1]), array([ 2, -2, -2]), 2]
[array([0, 0, 1]), array([ 2, -2, -3]), 1]
[array([1, 0, 1]), array([ 3, -2, -2]), 2]
[array([0, 0, 1]), array([ 3, -2, -3]), 1]
```

w矩阵的值为 [3 -2 -3]

b的值为 1

Process finished with exit code 0

```
误分类点  w  b
[array([0, 0, 0]), array([-1, -2, -2]), 1]
[array([1, 0, 0]), array([ 0, -2, -2]), 2]
[array([1, 0, 1]), array([ 1, -2, -1]), 3]
[array([0, 0, 1]), array([ 1, -2, -2]), 2]
```

```
[array([0, 0, 1]), array([ 1, -2, -3]), 1]
[array([1, 0, 1]), array([ 2, -2, -2]), 2]
[array([0, 0, 1]), array([ 2, -2, -3]), 1]
[array([1, 0, 1]), array([ 3, -2, -2]), 2]
[array([0, 0, 1]), array([ 3, -2, -3]), 1]
```

w矩阵的值为 [3 -2 -3]
b的值为 1

解向量为(3,-2,-3,1)

四. 部分代码解释

初始化

```
def __init__(self, x_train, y_train, eta = 1, w0 = 0, b0 = 0, alpha = 0):
    self.eta = eta
    self.b = b0
    #self.w = np.array([w0 for i in range(x_train.shape[1])])
    self.w = [-1, -2, -2]
    self.x_train = x_train
    self.y_train = y_train
    self.alpha = np.array([alpha for i in range(x_train.shape[0])])
    self.iter_matrix = []
```

初始化感知机模型的各种变量，其中w，b分别初始为（-1，-2，-2）和0，学习率初始为1，alpha是用于对偶算法的，这里没有用到。iter_matrix用于存储每次运行的结果。

学习算法

```
def orig_iter(self):
    adjust_flag = False
    iter_number = 0

    for x,y in zip(self.x_train, self.y_train):
        iter_number += 1

        while y * (np.dot(self.w,x) + self.b) <= 0:
            adjust_flag = True
            self.w += self.eta * y * x
            self.b += self.eta * y

            self.iter_matrix.append([x, copy.copy(self.w), self.b])

    if iter_number != 1 and adjust_flag:
        adjust_flag = False
        self.orig_iter()
```

adjust用于标记算法中是否有点分类错误，iter_number存储步数，对每个点进行判别，同时更新w和b,直至该点分类正确。

打印结果

```
def pprint(self):
    print('误分类点   w   b ')
    for i in self.iter_matrix:
        print(i)
```

运行过程

```
if __name__ == '__main__':
    data_x = [[0, 0, 0], [1, 0, 0], [1, 0, 1], [1, 1, 0], [0, 0, 1], [0, 1, 1], [0, 1, 0], [1, 1, 1]]
    data_y = [1, 1, 1, 1, -1, -1, -1, -1]
    data_x, data_y = np.array(data_x), np.array(data_y)

    a = Preceptron(data_x, data_y)
    a.orig_iter()
    a.pprint()
    print("-----")
    print("w矩阵的值为", a.iter_matrix[-1][1])
    print("b的值为", a.iter_matrix[-1][2])
```

By:184395-韩璐璐

2018/9/23