Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу «Нейроинформатика»

Студент: К. О. Вахрамян
Преподаватель: Н. П. Аносова
Группа: М8О-406Б

Дата:
Оценка:
Подпись:

Москва, 2021

Лабораторная работа №1

Задача:

Целью работы является исследование свойств персептрона Розенблатта и его применение для решения задачи распознавания образов.

Основные этапы работы :

1. Для первой обучающей выборки построить и обучить сеть, которая будет правильно относить точки к двум классам. Отобразить дискриминантную линию и проверить качество обучения.
2. Изменить обучающее множество так, чтобы классы стали линейно неразделимыми. Проверить возможности обучения по правилу Розенблатта.
3. Для второй обучающей выборки построить и обучить сеть, которая будет правильно относить точки к четырем классам. Отобразить дискриминантную линию и проверить качество обучения.

Вариант 2:

$$\begin{bmatrix} 2.6 & 3.6 & 0.1 & 0.8 & -3.1 & 2.4 \\ -3.4 & 4.8 & 3.8 & -3.5 & -1 & 3.2 \end{bmatrix} \quad \Big| \quad \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1.6 & 2.9 & 1.8 & -4.5 & -4.6 & 2.2 & 3.7 & -4.3 \\ 2.3 & 0.4 & 3.9 & -2 & -3.1 & 2.2 & 0.8 & 4.2 \end{bmatrix} \quad \Big| \quad \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

1 Описание

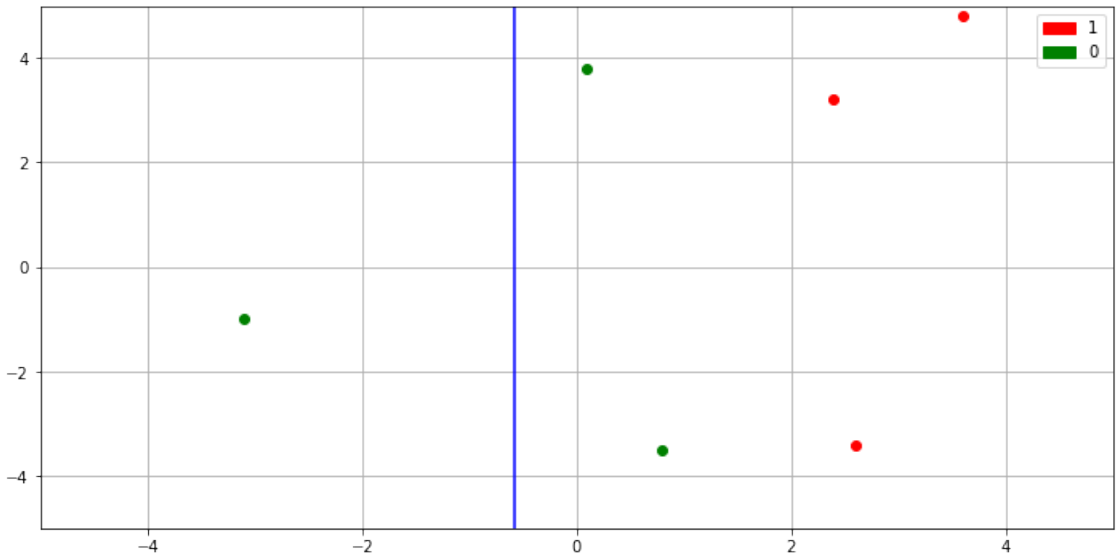
Класса слоя нейронов

```
class Perseptron:
    def __init__(self):
        self.w = None

    def Fit(self, X, Y, n_epoch=20, lr=0.1):
        P = np.ones((1, X.shape[1]))
        P = np.append(P, X, axis=0)
        S = Y.shape[0]
        R = P.shape[0]
        T = Y
        if self.w is None:
            self.w = np.random.rand(S, R)
        for epoch in range(n_epoch):
            for i in range(P.shape[1]):
                n = self.w * P[:,i]
                a = np.array([hardlim(x) for x in n])
                e = T[:,i] - a
                for j in range(S):
                    self.w[j] = self.w[j] + lr * e[j] * P[:,i].T
    def Predict(self, test, n):
        ans = np.zeros((n, test.shape[1]))

        P = np.ones((1, test.shape[1]))
        P = np.append(P, test, axis=0)

        for i in range(P.shape[1]):
            n = self.w  P[:,i]a = np.array([hardlim(x) for x in n])for j in
                range(a.shape[0]):ans[j][i] = a[j]return ans
```

Создаём модель для первой обучающей выборки

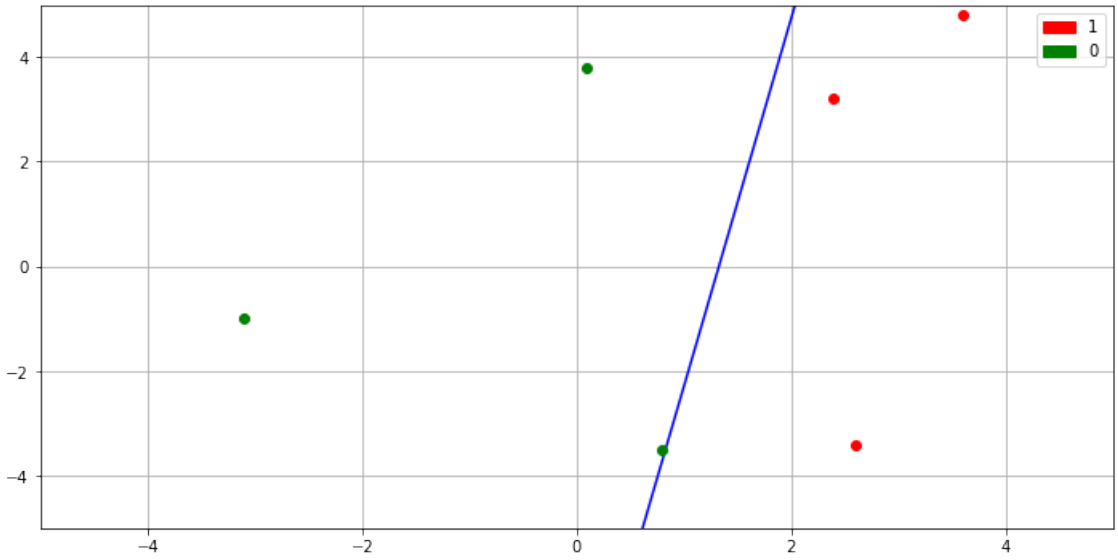
Модель со случайными весами:

```
net = Perseptron()
net.Fit(X, Y, n_epoch=0)
w = net.w
w
array([[4.17022005e-01, 7.20324493e-01, 1.14374817e-04]])
```



Модель после обучения 50 эпох

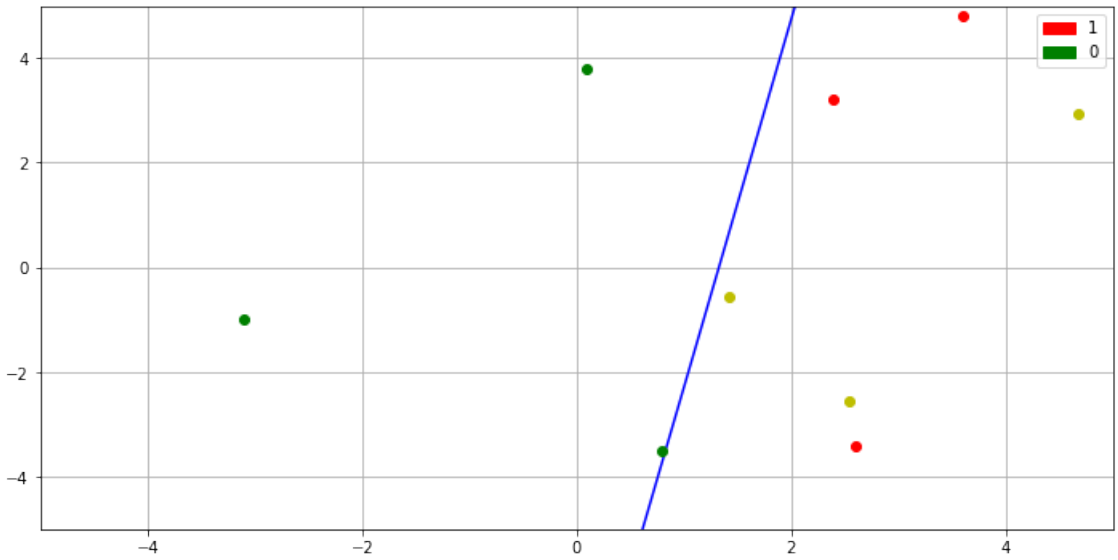
```
net = Perseptron()
net.Fit(X, Y, n_epoch=50)
w = net.w
w
array([[-1.09766743, 0.82675589, -0.11766141]])
```



Проверим модель на тестовой выборке

```
test = np.array([np.random.randint(-5, 5, 3) + np.random.random(3),
 np.random.randint(-5, 5, 3) + np.random.random(3)])
```

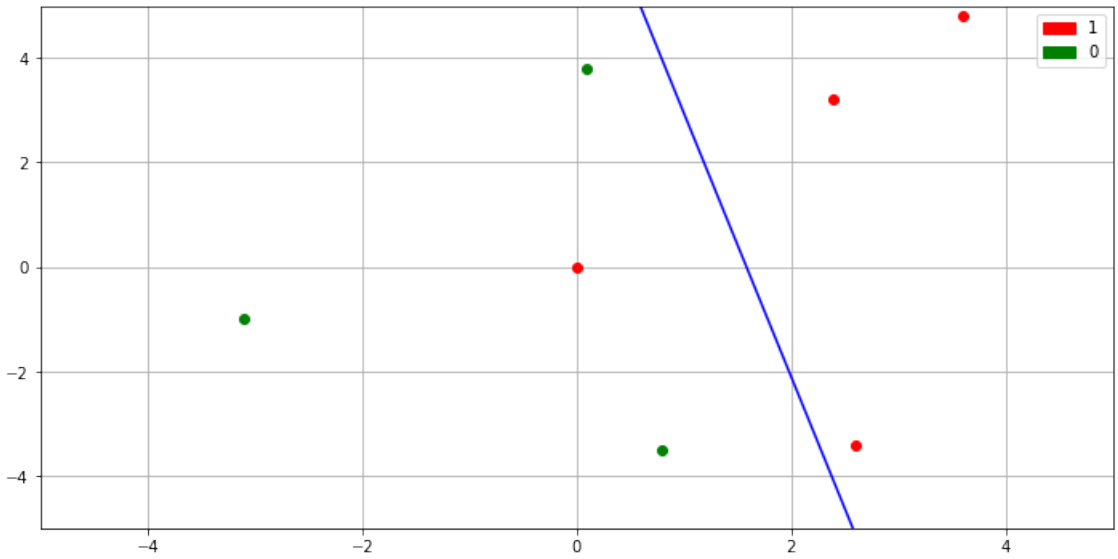
```
test
OneLayerPlt(X, Y, w, test)
net.Predict(test, 1)

array([[1., 1., 1.]])
```



Линейно неразделимая выборка

Обучение 50 эпох

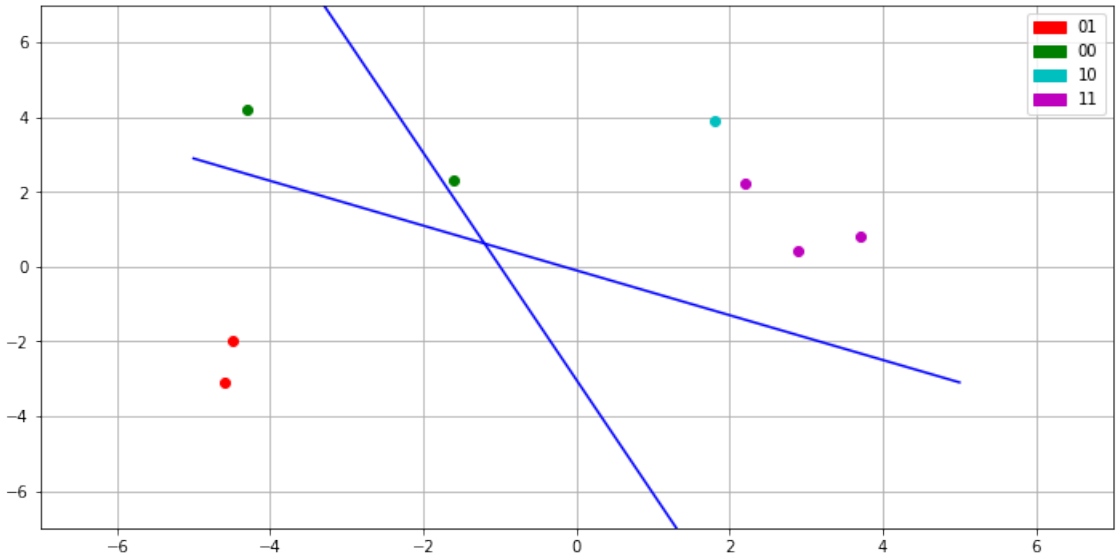
```
net = Perseptron()
net.Fit(X, Y, n_epoch=50)
w = net.w
w
array([[-0.42161076, 0.26597052, 0.0527575 ]])
```



Создаём модель для второй обучающей выборки

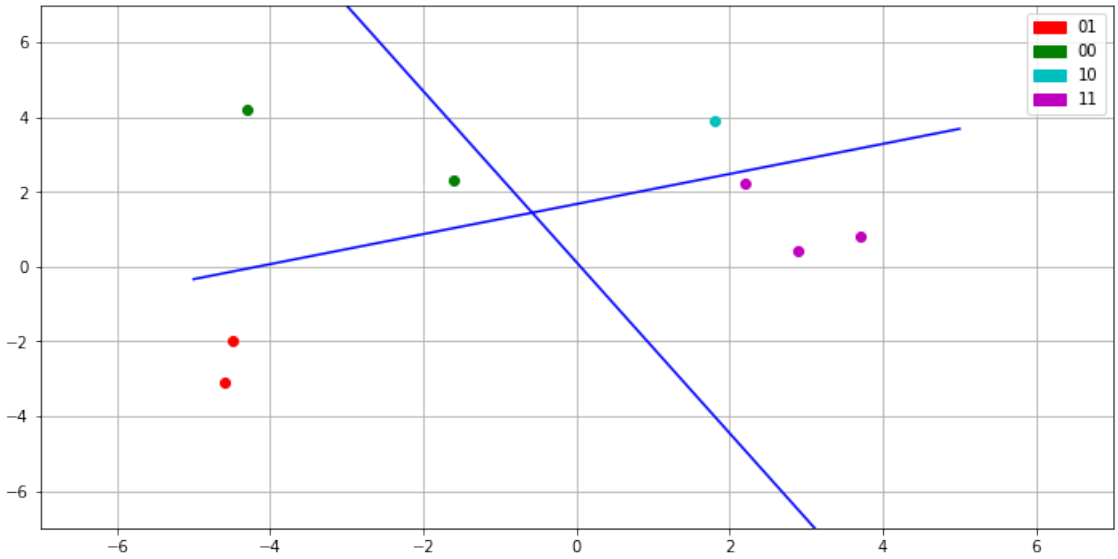
Модель со случайными весами:

```
net2 = Perseptron()
net2.Fit(X_2, Y_2, n_epoch=0)
w_2 = net2.w
w_2

array([[0.09280081, 0.51815255, 0.86502025],
       [0.82914691, 0.82960336, 0.27304997]])
```



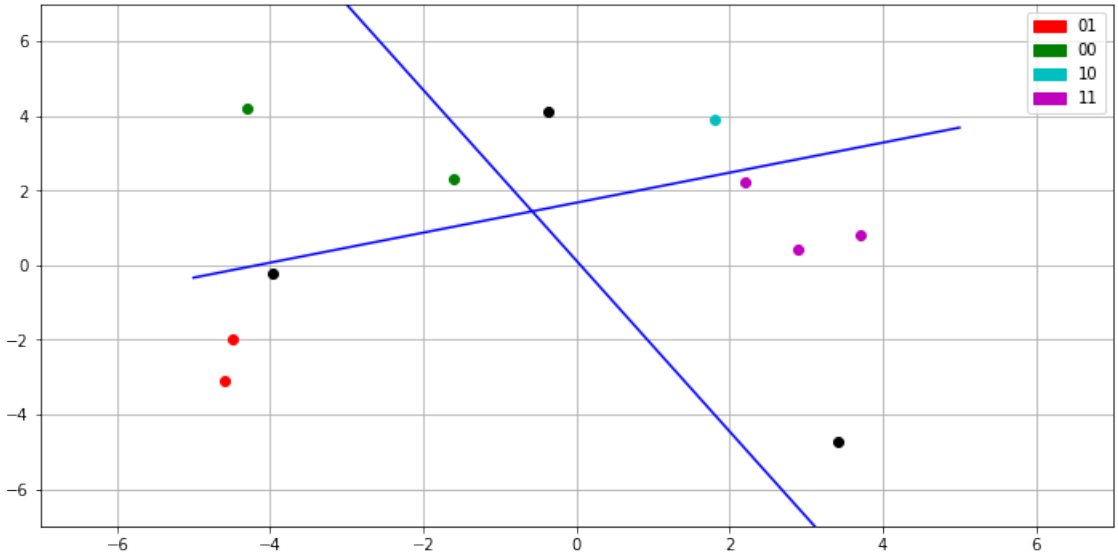
Модель после обучения 50 эпох

```
net2 = Perseptron()
net2.Fit(X_2, Y_2, n_epoch=50)
w_2 = net2.w
w_2

array([[-0.0407568 , 0.83052804, 0.36306552],
       [ 0.6716541 , 0.16178788, -0.4024491 ]])
```



Проверим модель на тестовой выборке

```
test_2 = np.array([np.random.randint(-5, 5, 3) + np.random.random(3),np.random.randint
    (-5, 5, 3) + np.random.random(3)])
MultyLayerPlt(X_2, Y_2, w_2, test_2)
a = net2.Predict(test_2,2)
print(a)

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


01
00
10
11
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6

2 Выводы

Выполнив первую лабораторную работу я реализовал персептрон Розенблатта.

Данная нейронная сеть имеет ряд недостатков, к примеру, только линейная классификация. Однако она очень проста в реализации и успешно выполняет поставленные задачи.