

**Московский авиационный институт**  
**(Национальный исследовательский университет)**  
Факультет прикладной математики и физики  
Кафедра вычислительной математики и программирования

**Лабораторная работа № 1**  
по курсу «Нейроинформатика»  
Тема: Персептроны. Процедура обучения  
Розенблатта.

Студент: Петросян В. А.  
Группа: 80-408Б-17  
Преподаватель: Аносова Н.П.  
Оценка:

Москва, 2020

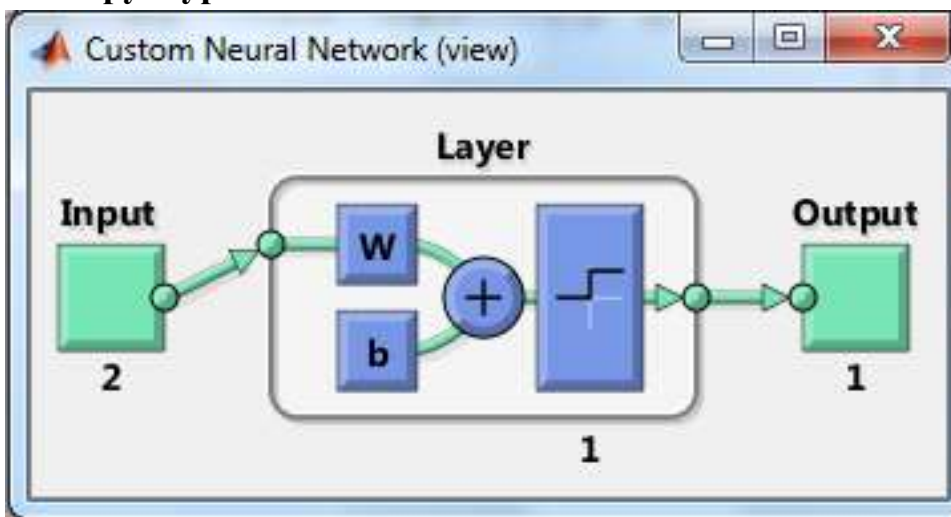
**Постановка задачи:** Исследование свойств персептрона Розенблатта и его применение для решения задачи распознавания образов.

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

### 1.1 Входные данные

$$15. \left| \begin{array}{cccccc} -4.1 & -1.7 & -3.7 & -4 & -0.1 & 2.1 \\ -2.4 & 1.7 & 2.2 & 1.5 & 2.7 & 4 \end{array} \right| \left| \begin{array}{cccccc} 1 & 1 & 0 & 0 & 1 & 1 \end{array} \right|$$

### 1.2 Структура сети



## 1.3 Алгоритма Розенблатта

```
Petrosyan_1_1.m  Rosenblatt.m  +
1  function net = Rosenblatt(net, P, T, iters, learnRate)
2  for j = 1:iters
3      for i = 1:6
4          p = P(:,i);
5          t = T(:,i);
6          b = net.b{1};
7          a = sim(net, p);
8          e = t - a;
9          if (mae(e))
10             net.IW{1,1} = net.IW{1,1} + p' * e * learnRate;
11             net.b{1} = b + 1 * e * learnRate;
12         end
13     end
14     disp(['On iteration : ', num2str(j)]);
15     disp('Error : ');
16     disp(mae(T - net(P)));
17     disp('Weights: ');
18     disp(net.IW{1,1});
19     disp('Biases : ');
20     disp(net.b{1});
21 end
22 end
```

### 1.3.1

Randomly generated Weights:

0.5844 0.9190

Randomly generated Biases :

0.8315

### 1.3.2

On iteration : 1

Error :

0.3333

Weights:

0.4344 0.3490

Biases :

1.1315

On iteration : 2

Error :

0

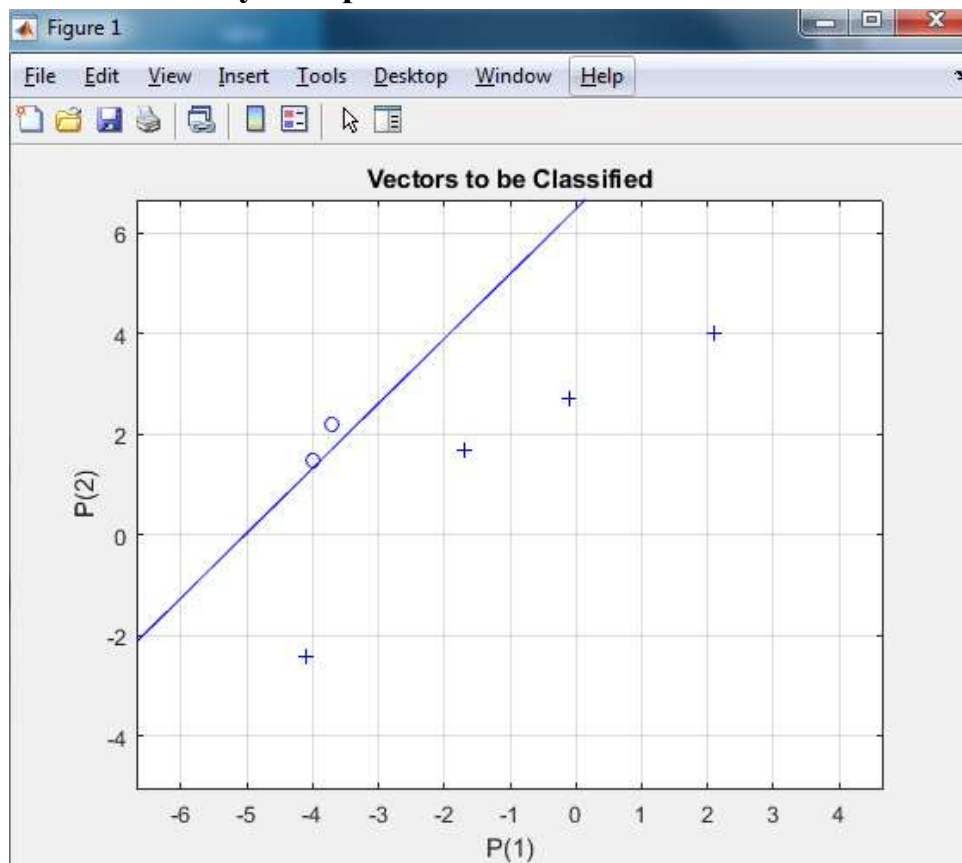
Weights:

0.2844 -0.2210

Biases :

1.4315

### 1.3.3 После двух итераций



### 1.4.1

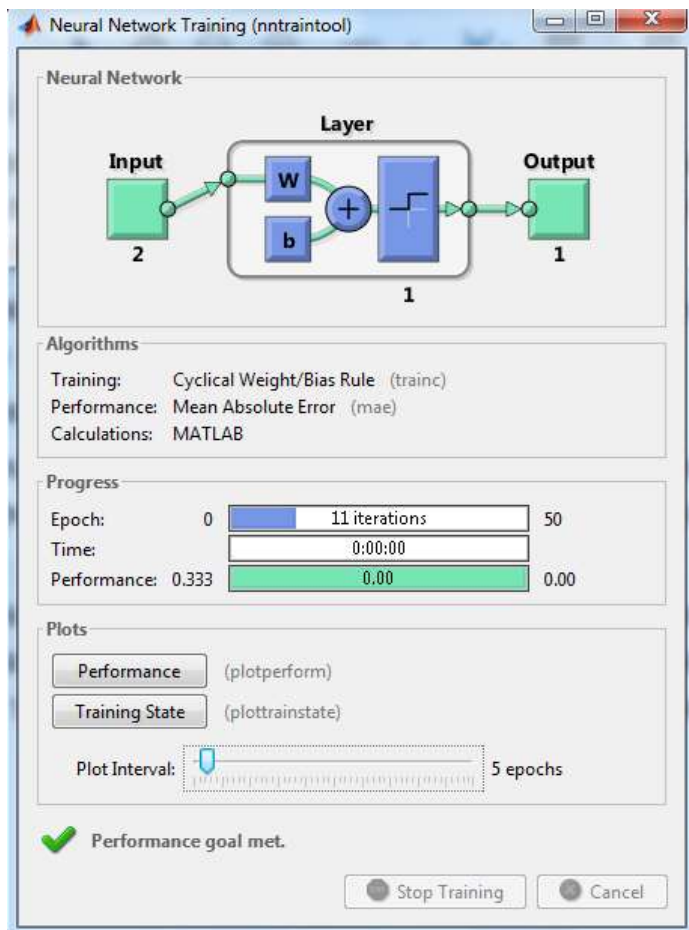
Randomly generated Weights:

-0.9286 0.6983

Randomly generated Biases :

0.3115

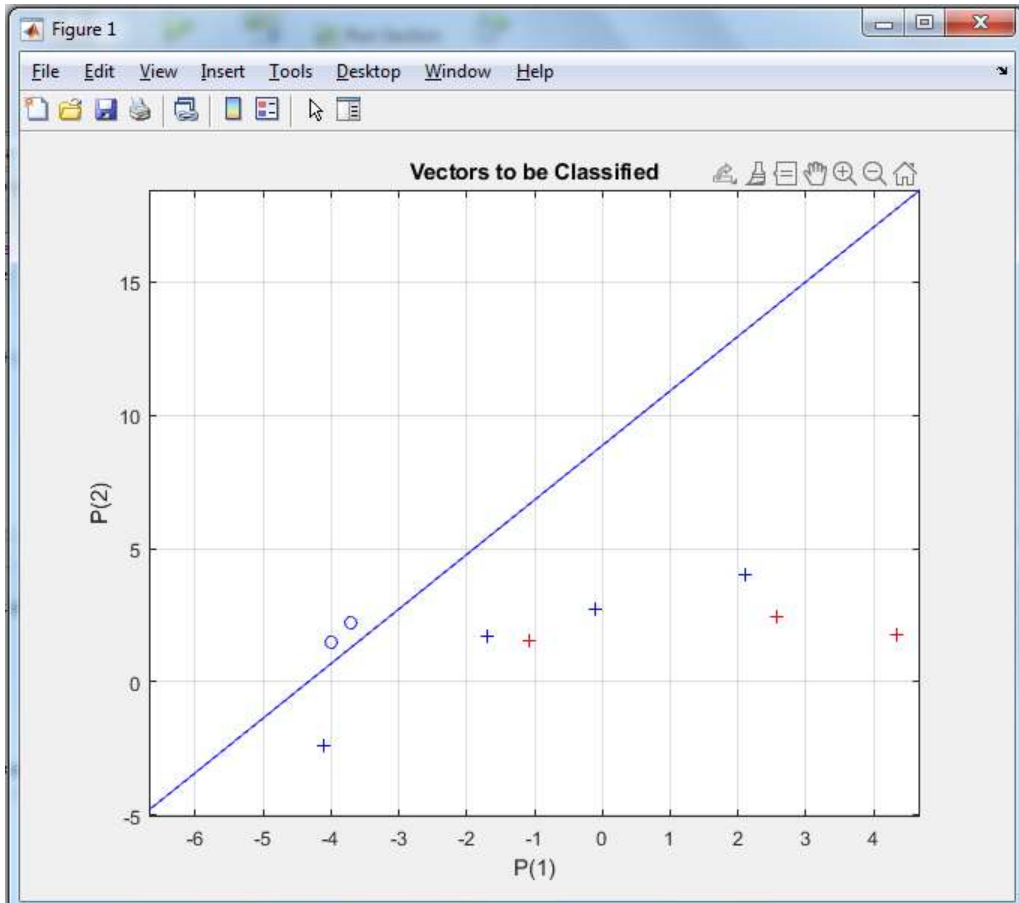
## 1.4.2



After 50 iters Weights:  
3.0714 -1.5017

After 50 iters Biases :  
13.3115

### 1.4.3



2.1 В качестве неразделяемых данных сделал обычный хор.

### 2.2

Randomly generated Weights:

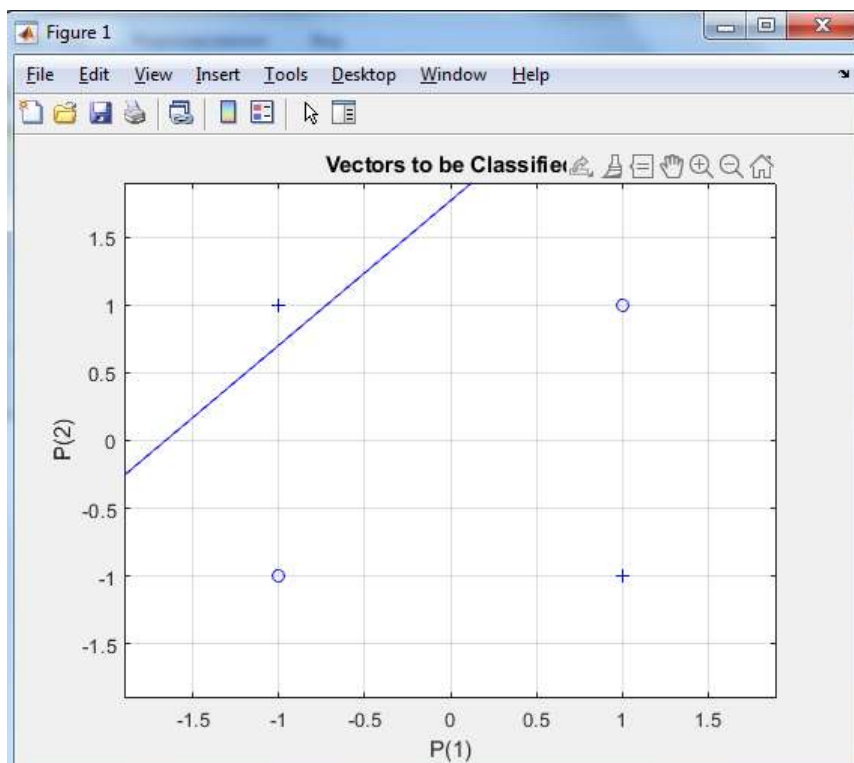
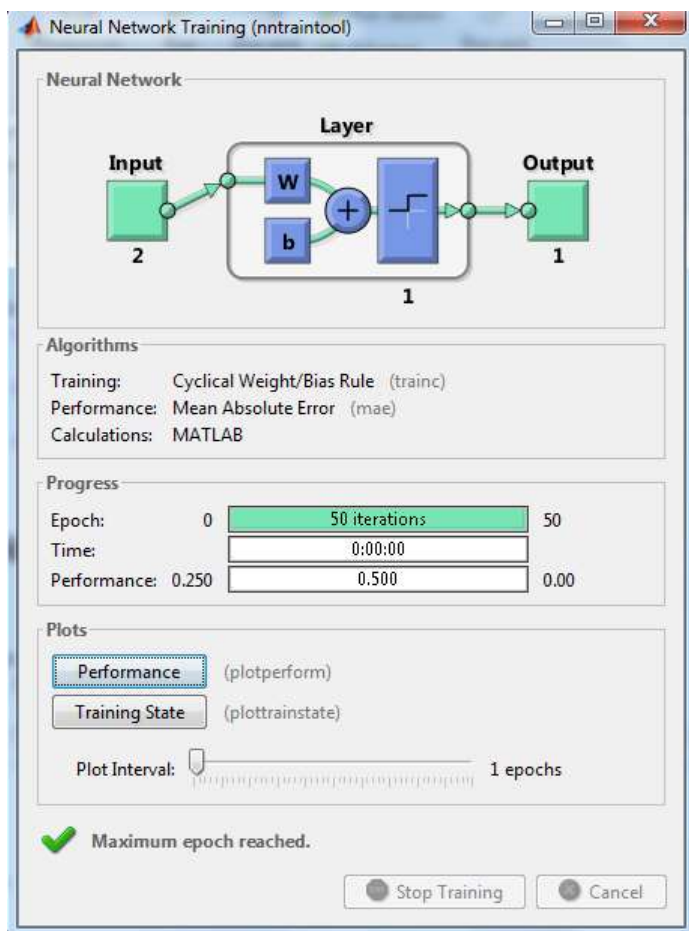
0.3897 -0.3658

Randomly generated Biases :

0.6469

### 2.3

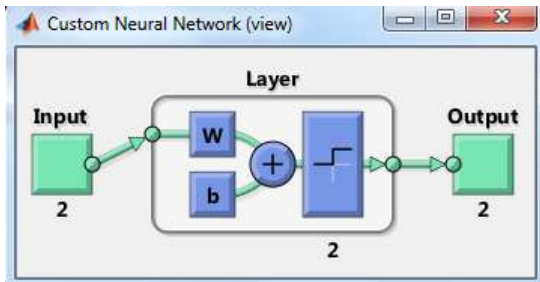
ans = 0.7500



3.1

$$\begin{bmatrix} 2 & -2.3 & -4.1 & 1.9 & 4.5 & -0.7 & 2.6 & -3.2 \\ -4.7 & -4.6 & 3.2 & -1.9 & -4.7 & -1.2 & 2.9 & -0.2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

3.2



3.3

Randomly generated Weights:

-0.1225   0.5310

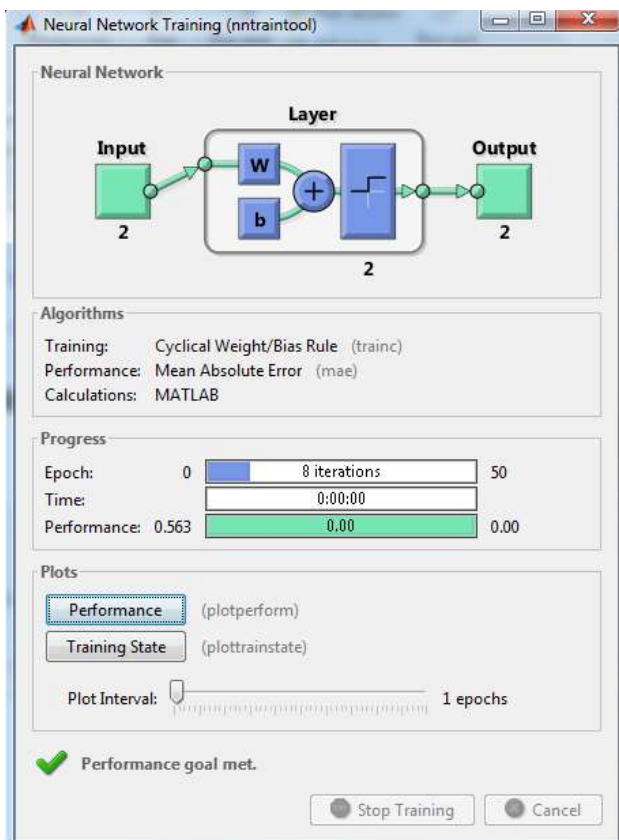
-0.2369   0.5904

Randomly generated Biases :

0.9004

-0.9311

3.4





Weights:

-1.3225 -4.3690

-5.6369 -0.4096

Biases :

2.9004

-4.9311

