

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

"Уфимский государственный авиационный технический университет"

Кафедра Высокопроизводительных вычислительных технологий и систем

Дисциплина: Интеллектуальные системы

Отчет по практической работе № 2

Тема: «Персептрон Розенблатта для распознавания цифр»

Группа ПМ-453	Фамилия И.О.	Подпись	Дата	Оценка
Студент	Шамаев И.Р.			
Принял	Казакова Т.Г.			

Уфа 2022

В качестве примера линейной разделимости нескольких классов объектов рассмотрим задачу распознавания арабских цифр. Каждую цифру будем изображать в простейшей пиксельной форме размерностью 5x3 и кодировать 15-мерным вектором, используя построчный обход.

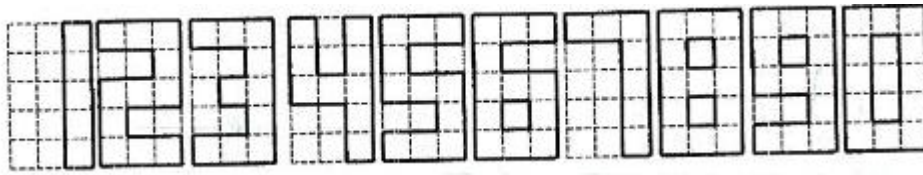


Рисунок 1. Образы цифр, используемые для обучения

Пусть каждая цифра имеет только одно изображение, показанное на рис. 1, тогда в каждом классе содержится один объект. Геометрически каждую цифру можно сопоставить с вершиной единичного гиперкуба в 15-мерном пространстве. Очевидно, что каждая вершина может быть отделена от остальных вершин гиперплоскостью, поэтому задача распознавания цифр в данной постановке является линейно разделимой.

Обучение сети осуществляется по эпохам. Эпоха содержит 10 примеров по числу распознаваемых цифр от 0 до 9. Внутри примера цифры могут чередоваться детерминированным или случайным образом. Каждый пример содержит векторы X и D . Вектор X (16-мерный) подается на основной вход сети, вектор D (10-мерный) подается на обучающий вход сети.

Программа

```
clear all;

перох=20; %(Число эпох)

h=1; %(Параметр обучения)

n=7; %(Коэффициенты этого нейрона выводятся на график)

x=[1 -1 -1 1 -1 -1 1 -1 -1 1 -1 -1 1 -1 1; 1 1 1 1 -1 -1 1 1 1 1 1 -1 -1 1 1;
   1 1 1 1 -1 -1 1 -1 1 1 1 -1 -1 1 1; 1 1 -1 1 1 -1 1 1 1 1 -1 -1 1 -1 1;
   1 1 1 1 1 -1 -1 1 1 1 -1 -1 1 1 1; 1 1 1 1 1 -1 -1 1 1 1 1 -1 1 1 1;
   1 1 1 1 -1 -1 1 -1 -1 1 -1 -1 1 -1 1; 1 1 1 1 1 -1 1 1 1 1 1 -1 1 1 1;
   1 1 1 1 1 -1 1 1 1 1 -1 -1 1 1 1; 1 1 1 1 1 -1 1 1 -1 1 1 -1 1 1 1];

w=rand(10,16)-0.5; %(Матрица начальных весовых коэффициентов)
```

```

for k=1:непox %(Обучение по эпохам)

    e2(k)=0;

    for m=1:10 %(Обучение по примерам)

        d=zeros(1,10);

        d(m)=1; %(Обучающий сигнал)

        for j=1:16

            xp(j)=x(m,j);

        end

        u=w*xp';

        for i=1:10

            if u(i)>=0 y(i)=1; else y(i)=0; end; %(Функция активации)

        end

        e=d-y; %(Вектор-строка ошибок обучения)

        w=w+h*e'*xp; %(Алгоритм обучения в векторно-матр. форме)

        e2(k)=e2(k)+e*e'; %(Суммарный квадрат ошибок обучения)

        nit=(k-1)*10+m; %(Номер текущей итерации)

        wlg(1,nit)=w(n,1); wlg(2,nit)=w(n,2); wlg(3,nit)=w(n,16);

    end

end

subplot(2,1,1);

hold on;

plot(wlg(1,:), 'r');

plot(wlg(2,:), 'b');

plot(wlg(3,:), 'g');

grid on;

```

```
subplot(2,1,2);
```

```
plot(e2,'b');
```

```
grid on;
```

Результат

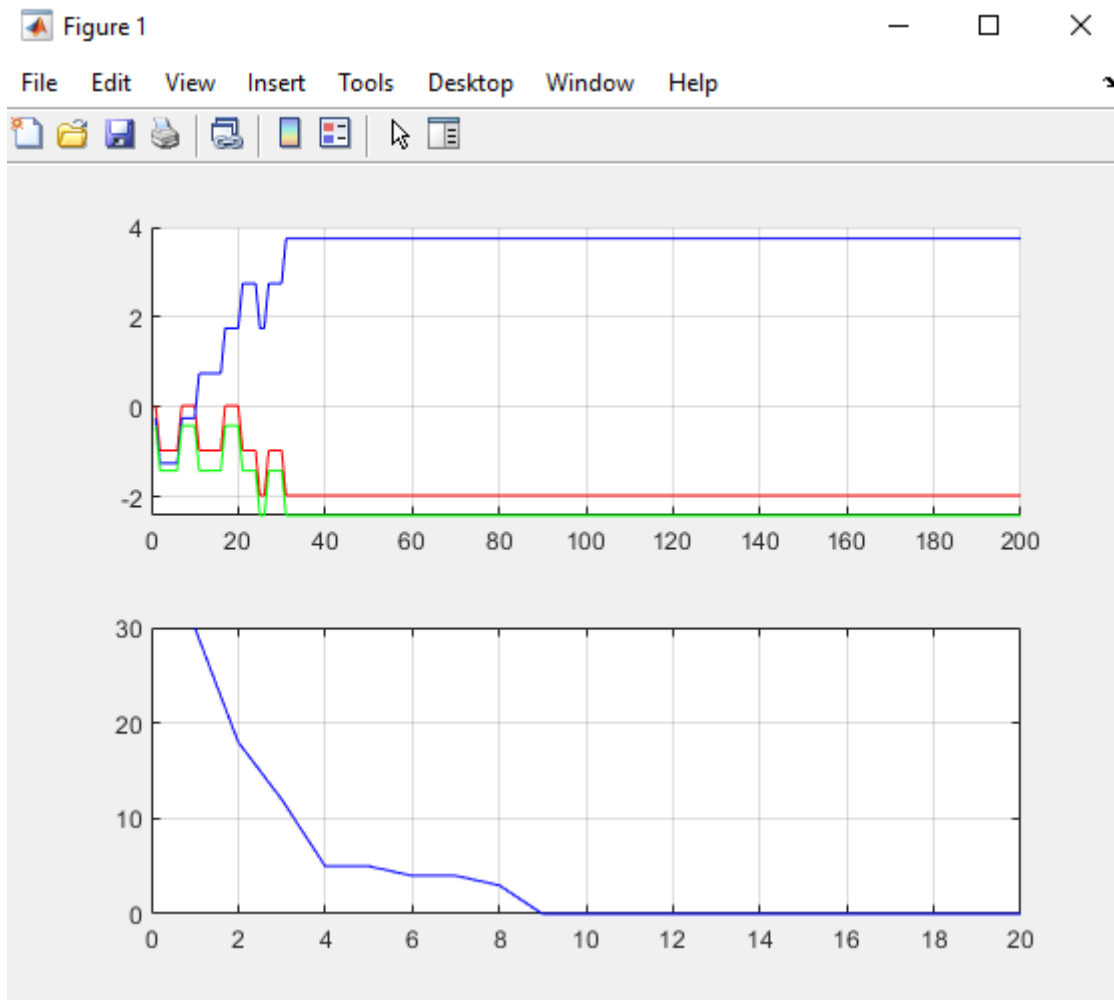


Рисунок 2. Пример реализации программы

В процессе моделирования получались различные реализации процедуры обучения персептрона, в которых начальные весовые коэффициенты задавались случайным образом. Естественно, что обучение заканчивалось за разное число эпох, в зависимости от расположения начальных весовых коэффициентов к границам раздела. На рис. 2 приведена одна из реализаций. На верхнем графике показан характерный процесс коррекции весовых коэффициентов. На нижнем графике показано изменение

суммарного квадрата ошибки. В данной реализации обучения сети состоялось за 9 эпох