### Аппроксимация данных

1. Первым делом выполняем загрузку данных:

|  |
| --- |
| X = -3.1:.1:3.1;  T = load('P10.mat')  T1=T.P10(1,:)  plot(X,T1,'+');  title('Training Vectors');  xlabel('Input Vector P');  ylabel('Target Vector T');     1. Обучаем нашу сеть, взяв распространение радиальных базовых нейронов слишком маленьким. Получаем эффект Underlapping Neurons.   eg = 0.00001;  sc = 0.0000001; %spread of the radial basis  net = newrb(X,T1,eg,sc);  plot(X,T1,'+');  xlabel('Input');  X = -3.1:.1:3.1;  Y = net(X);  hold on;  plot(X,Y);  hold off;  legend({'Target','Output'})  pause |
| Структура сети представлена ниже |

Как мы видим, в сети 59 нейронов

1. Обучаем нашу сеть, взяв распространение радиальных базовых нейронов слишком маленьким. Получаем эффект Overlapping Neurons.  
     
   eg = 0.00001;

sc = 2; %spread of the radial basis

net = newrb(X,T1,eg,sc);

plot(X,T1,'+');

xlabel('Input');

X = -3.1:.1:3.1;

Y = net(X);

hold on;

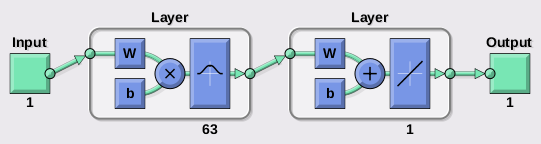
plot(X,Y);

hold off;

legend({'Target','Output'})

pause

|  |
| --- |
| X = -3.1:.1:3.Pasdasd10.mat') plot(X,T1,'+'); |

Структура сети представлена ниже   


Как мы видим, в сети 64 нейрона

1. Вывод: мы заметили, что если распространение радиальных базовых функций spread слишком мало, то происходит эффект Underlapping Neurons, то есть сеть требует слишком много нейронов. Если же, наоборот, spread слишком велико, то происходит эффект overlapping Neurons и сеть не может обучиться.

### Классификация данных

1. Произведем загрузку исходных данных и выведем их на плоскость:  
   %Исходные данные

P = [1.3 1.2; 0.26 0.25; 0.24 1.2; 1.3 0.25;

0.50 0.5; 1 0.5; 1 1; 0.15 0.5;

0.7 0.8; 0.9 0.6; 0.6 0.8; 0.7 0.8]';

Tc = [1 1 1 1 2 2 2 2 3 3 3 3];

%Вычислим размер Р (пригодится далее)

l = length(P)

%Вывод классов на плоскость

plot(P(1,1:4),P(2,1:4),'.','markersize',30,'color', 'g')

hold on

plot(P(1,9:12),P(2,9:12),'.','markersize',30,'color', 'b')

hold on

plot(P(1,5:8),P(2,5:8),'.','markersize',30,'color', 'r')

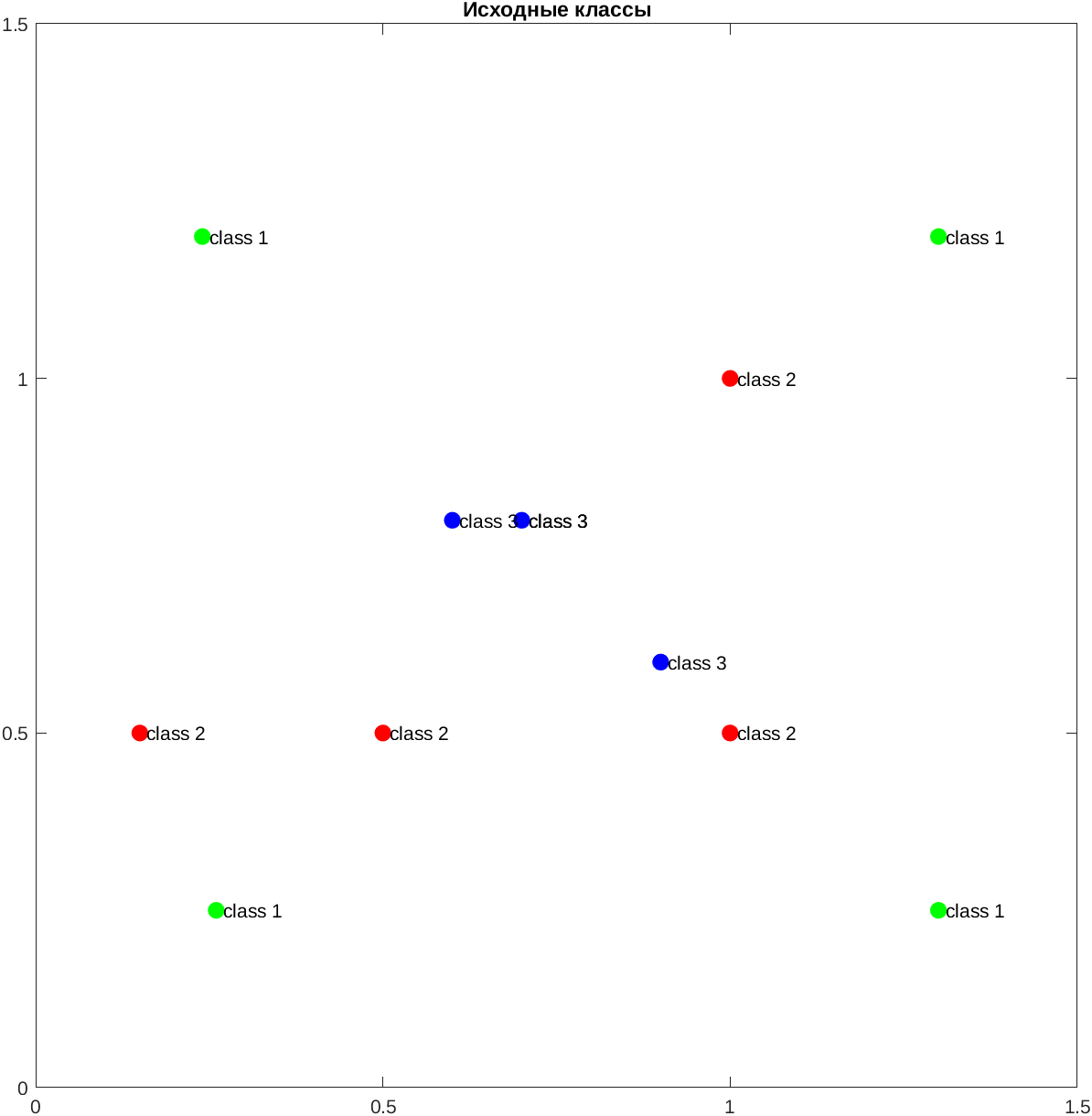
for i = 1:l, text(P(1,i)+0.01,P(2,i),sprintf('class %g',Tc(i))), end

title('Исходные классы')

axis([0 1.5 0 1.5])

pause

hold off



1. Обучим нашу сеть и проверим ее на сначала на исходных данных, а потом добавим новую точку с координатами (1.5;1):

%Создаем нейронную сеть

T= ind2vec(Tc)

spread = 0.1 %Коэффициент, который влияет на "степень" обучения

% Если он будет слшиком большой то сеть не обучиться, слишком маленький

% Переобучиться и создат для каждой точки свой класс

net = newpnn(P, T,spread)

%Тестируем сеть на исходных данных

Y = net(P)

Yc = vec2ind(Y)

plot(P(1,1:4),P(2,1:4),'.','markersize',30,'color', 'g')

hold on

plot(P(1,9:12),P(2,9:12),'.','markersize',30,'color', 'b')

hold on

plot(P(1,5:8),P(2,5:8),'.','markersize',30,'color', 'r')

axis([0 1.5 0 1.5])

for i = 1:l,text(P(1,i)+0.01,P(2,i),sprintf('class %g',Yc(i))),end

title('Тестирование нейросети')

xlabel('P(1,:)')

ylabel('P(2,:)')

pause

%Тестируем сеть на новых данных (векторе Х)

x=[1.5;1];

y = net(x)

yc = vec2ind(y)

hold on

plot(x(1),x(2),'.','markersize',30,'color',[0 0 0])

text(x(1)+0.01,x(2),sprintf('class %g',yc))

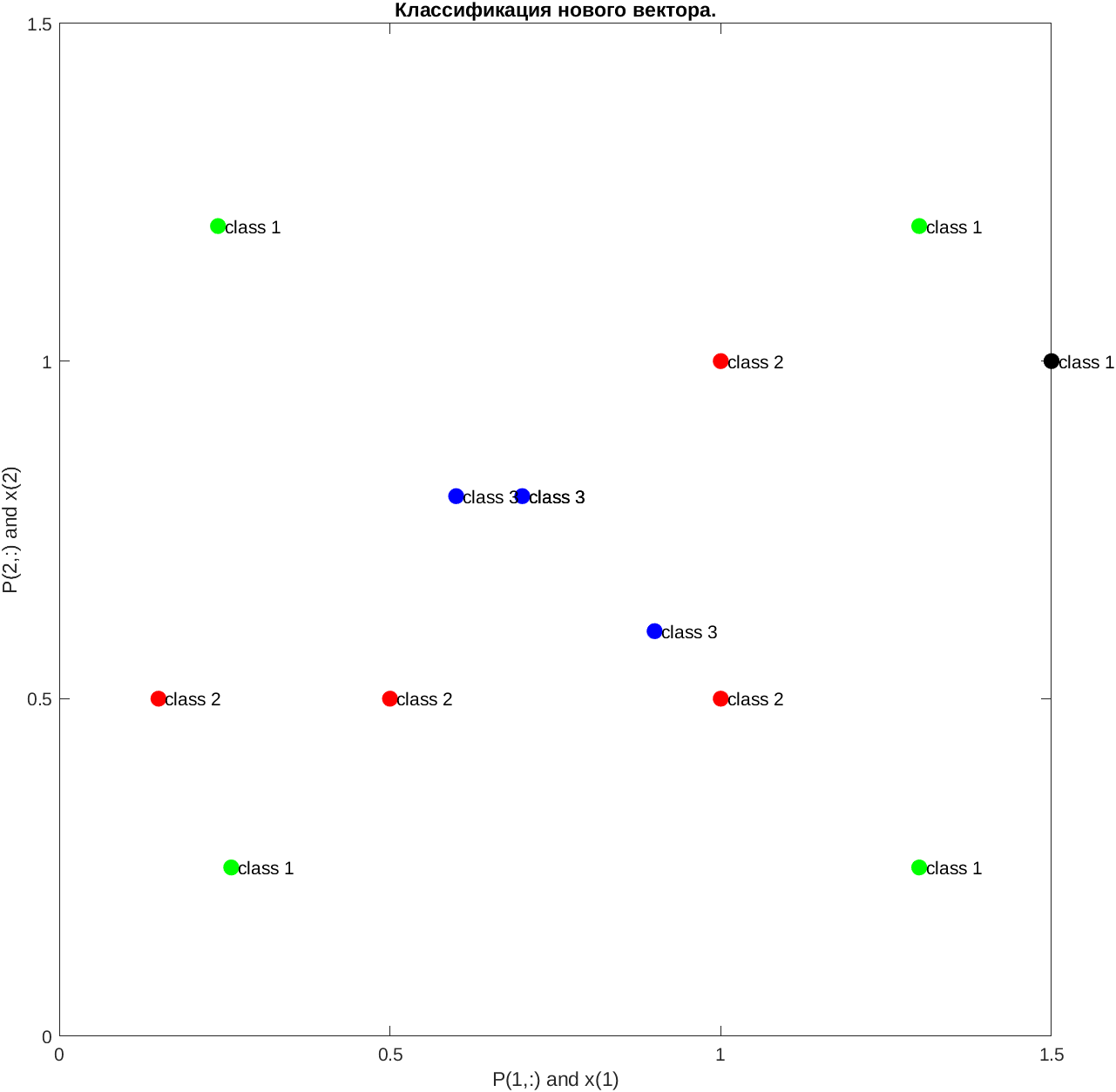
hold off

title('Классификация нового вектора.')

xlabel('P(1,:) and x(1)')

ylabel('P(2,:) and x(2)')

pause



1. Выведем область классификации

%Постройка областей классификации

x1 = 0:.05:2;

x2 = x1;

[X1,X2] = meshgrid(x1,x2);

xx = [X1(:) X2(:)]';

yy = net(xx);

yy = full(yy);

m = mesh(X1,X2,reshape(yy(1,:),length(x1),length(x2)));

m.FaceColor = [0 0.5 1];

m.LineStyle = 'None'

hold on

m = mesh(X1,X2,reshape(yy(2,:),length(x1),length(x2)));

m.FaceColor = [0 1.0 0.5];

m.LineStyle = 'None'

m = mesh(X1,X2,reshape(yy(3,:),length(x1),length(x2)));

m.FaceColor = [0.5 0 1];

m.LineStyle = 'None'

plot3(P(1,1:4),P(2,1:4),[1 1 1 1 ]+0.1,'.','markersize',30,'color', 'g')

plot3(P(1,9:12),P(2,9:12),[1 1 1 1 ] +0.1,'.','markersize',30,'color', 'b')

plot3(P(1,5:8),P(2,5:8),[1 1 1 1 ]+ 0.1,'.','markersize',30,'color', 'r')

for i = 1:l, text(P(1,i)+0.01,P(2,i),1,sprintf('class %g',Tc(i))), end

pause

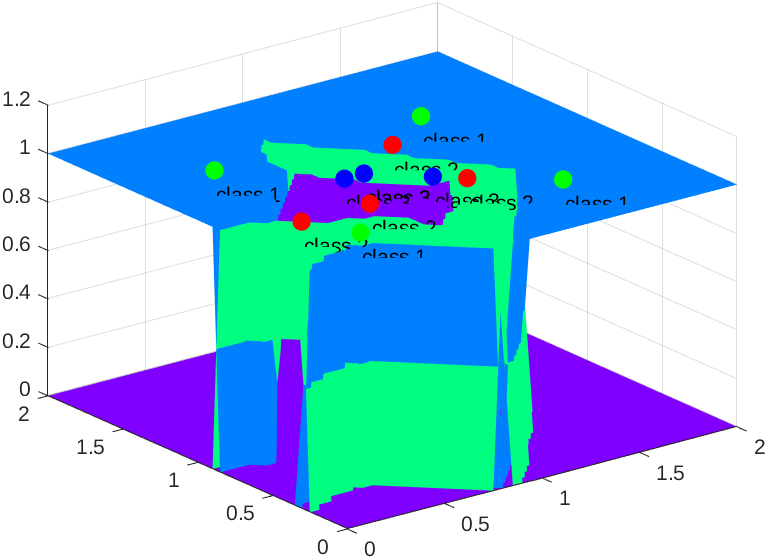
hold off

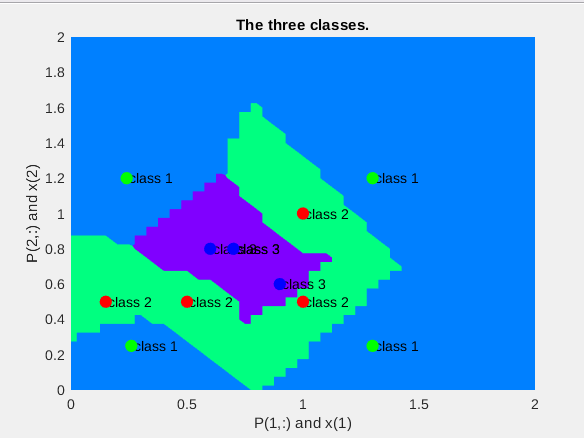
view(2)%"Сжимает все на плоскость"

title('The three classes.')

xlabel('P(1,:) and x(1)')

ylabel('P(2,:) and x(2)')





Была построена РБФ сеть для аппроксимации данных с помощью функции newrb. Рассмотрен эффекты «Underlapping Neurons» и

«Overlapping Neurons», которые могут возникнуть при неправильном выборе распространения радиальных базовых функций.

Во второй части данной работы были классифицированы точки на плоскости с помощью РБФ сети, созданной функцией newpnn.