

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

**Лабораторная работа № 4**  
по курсу «Нейроинформатика»  
Тема: Сети с радиальными базисными элементами.

Студент: Ваньков Д. А.  
Группа: 80-407Б-17  
Преподаватель: Аносова Н.П.

Москва, 2021

## Постановка задачи

Исследование свойств некоторых видов сетей с радиальными базисными элементами, алгоритмов обучения, а также применение таких сетей в задачах классификации и аппроксимаций функций.

1. Использовать вероятностную нейронную сеть для классификации точек в случае, когда классы не являются линейно разделимыми.
2. Использовать сеть с радиальными базисными элементами для классификации точек в случае, когда классы не являются линейно разделимыми.
3. Использовать обобщенно-регрессионную нейронную сеть для аппроксимации функции. Проверить работу сетей с рыхлыми данными.

Вариант 6

- |    |  |
|----|--|
| 6. | Эллипс: $a = 0.4, b = 0.15, \alpha = \pi/6, x_0 = 0.1, y_0 = -0.15$<br>Эллипс: $a = 0.7, b = 0.5, \alpha = -\pi/3, x_0 = 0, y_0 = 0$<br>Эллипс: $a = 1, b = 1, \alpha = 0, x_0 = 0, y_0 = 0$ |
|----|--|

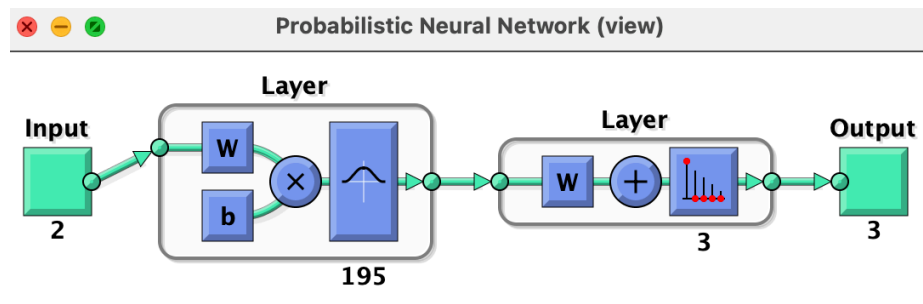
$$x = \sin(0.5t^2 - 5t), \quad t \in [0, 2], \quad h = 0.01$$

*traincgp, trainlm*

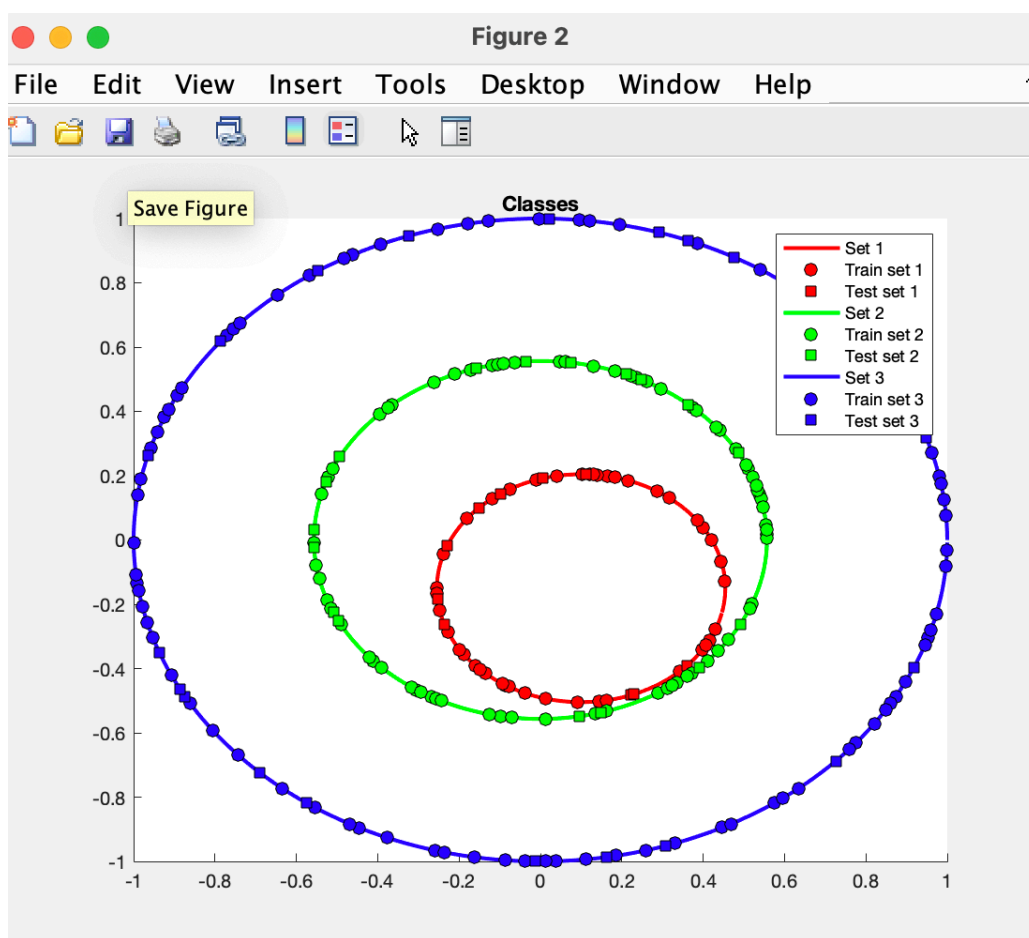
## Ход работы

1.

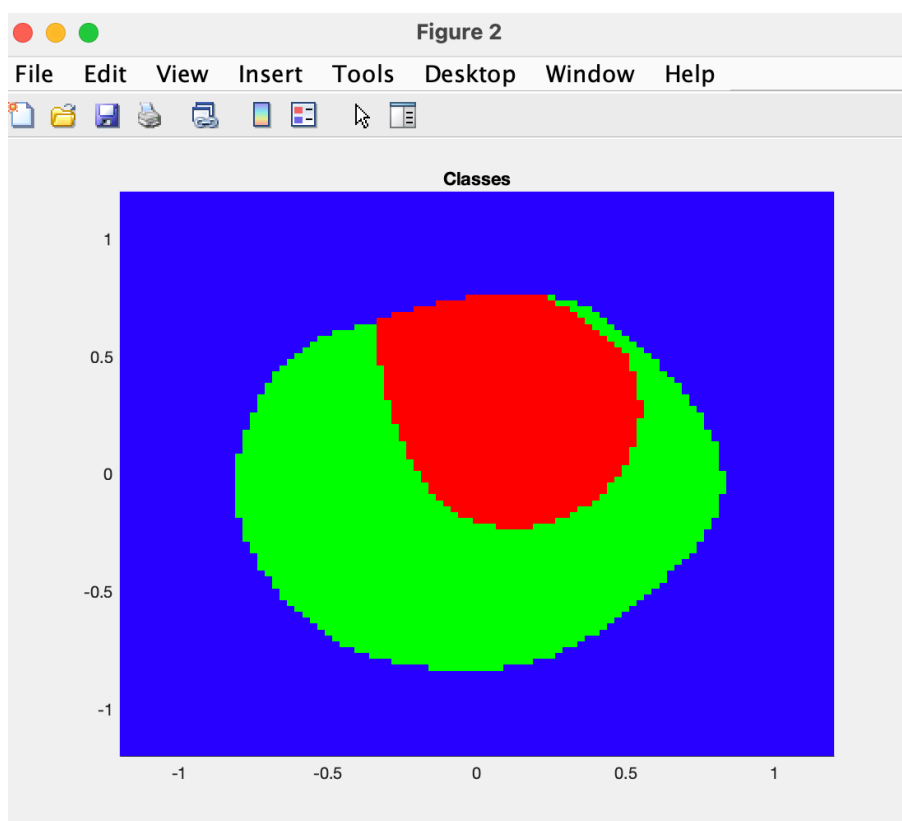
Структура сети



## Исходные множества



Конечный результат для вероятностной сети при  $\text{Spread} = 0.3$



Конечный результат для вероятностной сети при Spread = 0.1

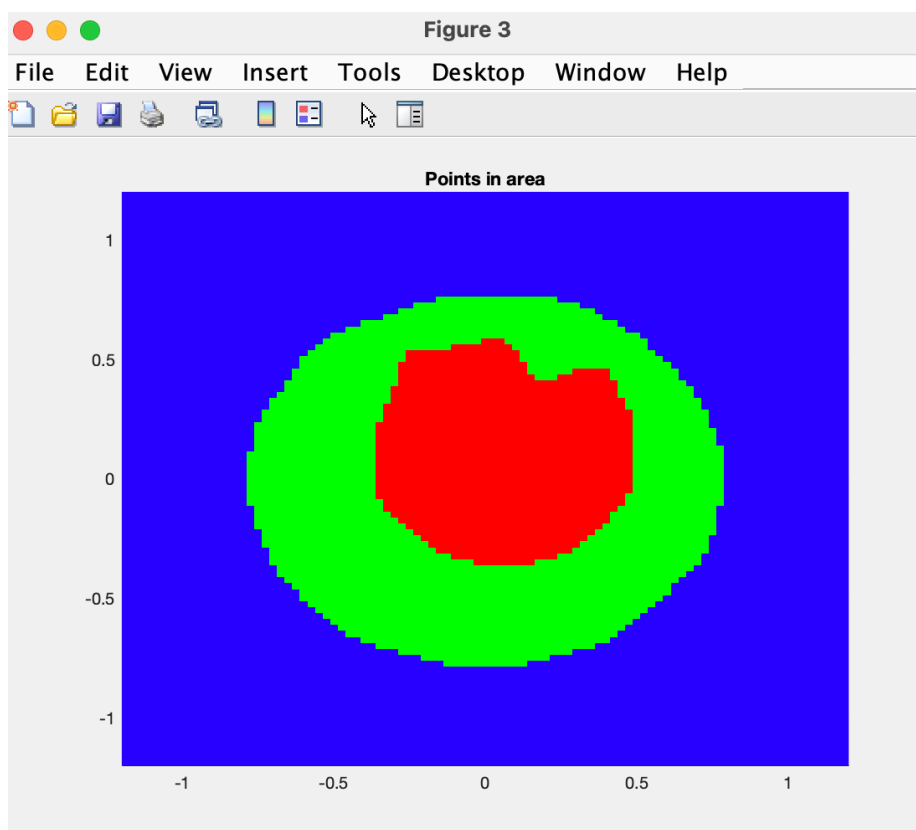


Таблица 1.

Функция создания сети	newpnn
Входной слой	2
Скрытый слой	195
Выходной слой	3
Активационные функции	radbas, compet
Динамика задержки	
Число примеров в подмножествах	195
Метод обучения	train
Параметры обучения	spread=0.3
Метод инициализации сети	initlay
Критерий окончания обучения	
Причина окончания обучения	
Число эпох обучения	

Train size: 195

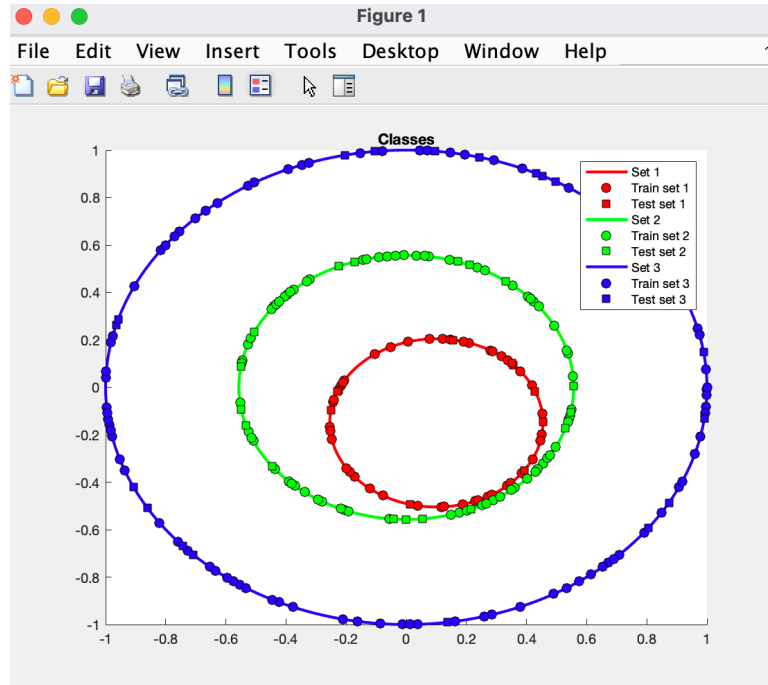
Test size: 48

Matches: 177

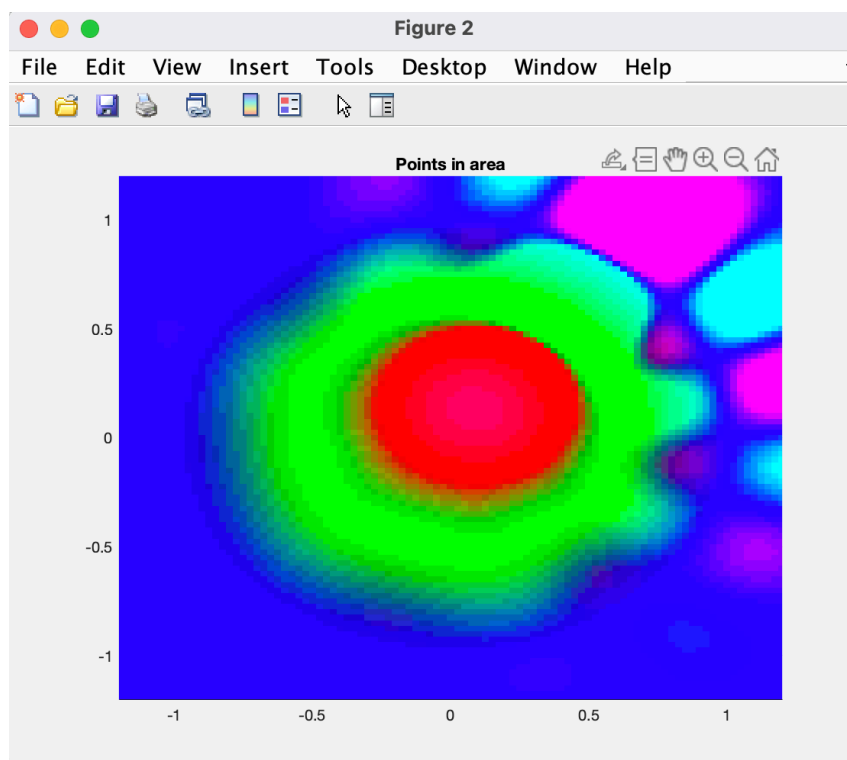
Matches: 42

2.

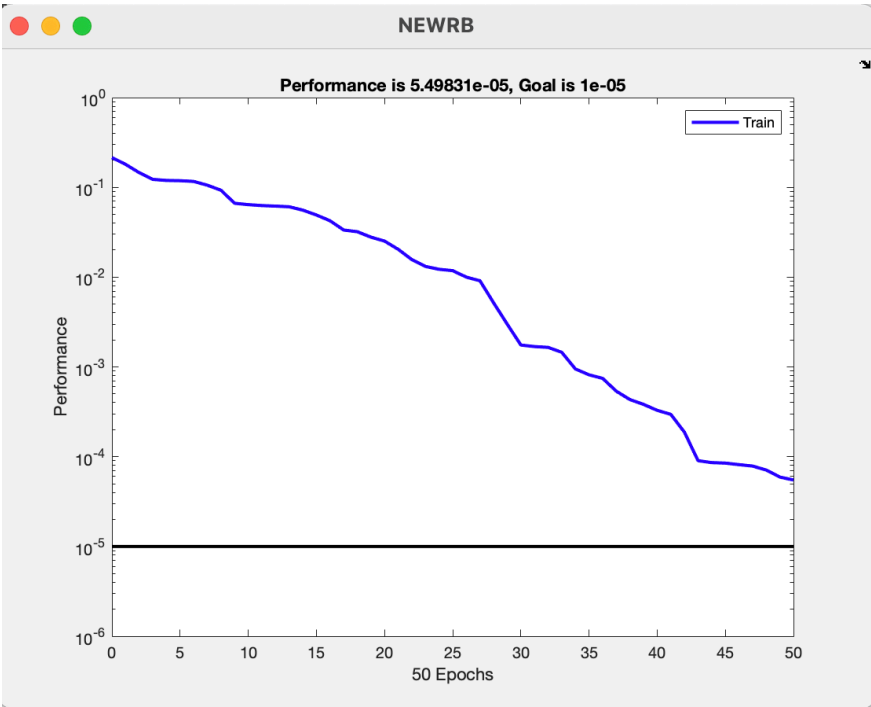
Исходные множества



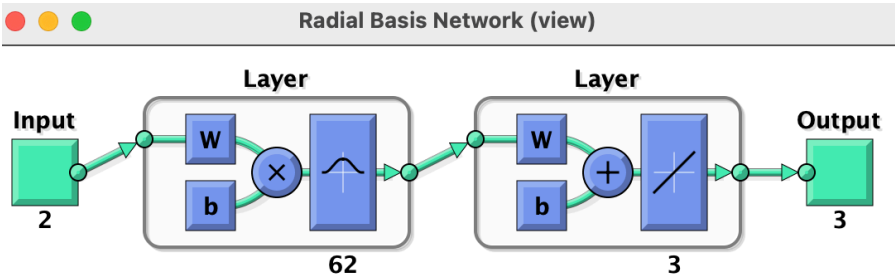
Конечный результат для сети с радиальными базисными элементами при Spread = 0.3



Performance

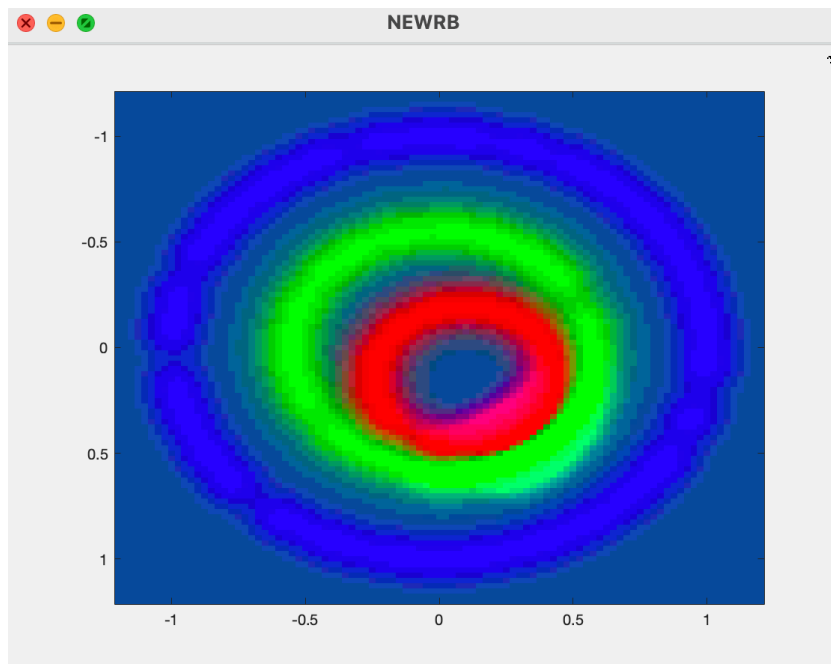


Структура сети и Таблица 1.

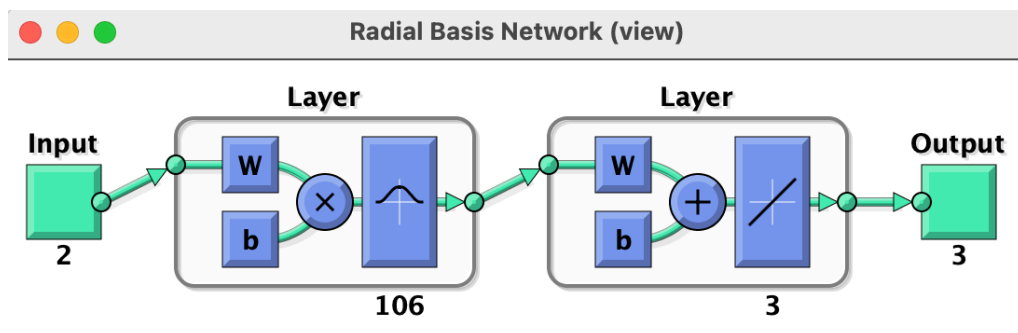


Функция создания сети	newrb
Входной слой	2
Скрытый слой	62
Выходной слой	3
Активационные функции	radbas, purelin
Динамика задержки	
Число примеров в подмножествах	195
Метод обучения	train
Параметры обучения	spread=0.3
Метод инициализации сети	initlay
Критерий окончания обучения	goal=1e-5
Причина окончания обучения	Максимальное количество эпох
Число эпох обучения	50

Конечный результат для сети с радиальными базисными элементами при Spread = 0.1



Структура сети



NEWRB, neurons = 0, MSE = 0.214526

NEWRB, neurons = 50, MSE = 4.23586e-05

Train size: 195

Test size: 48

Matches: 0

Matches: 0

NEWRB, neurons = 0, MSE = 0.214526

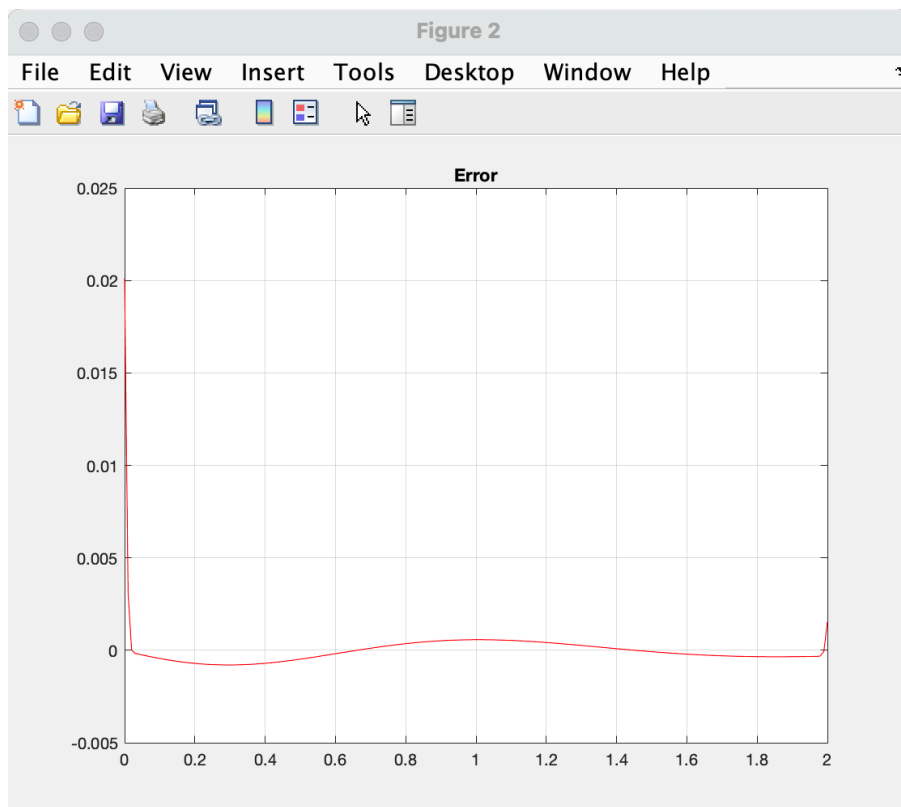
NEWRB, neurons = 50, MSE = 0.00892191

NEWRB, neurons = 100, MSE = 4.05871e-05

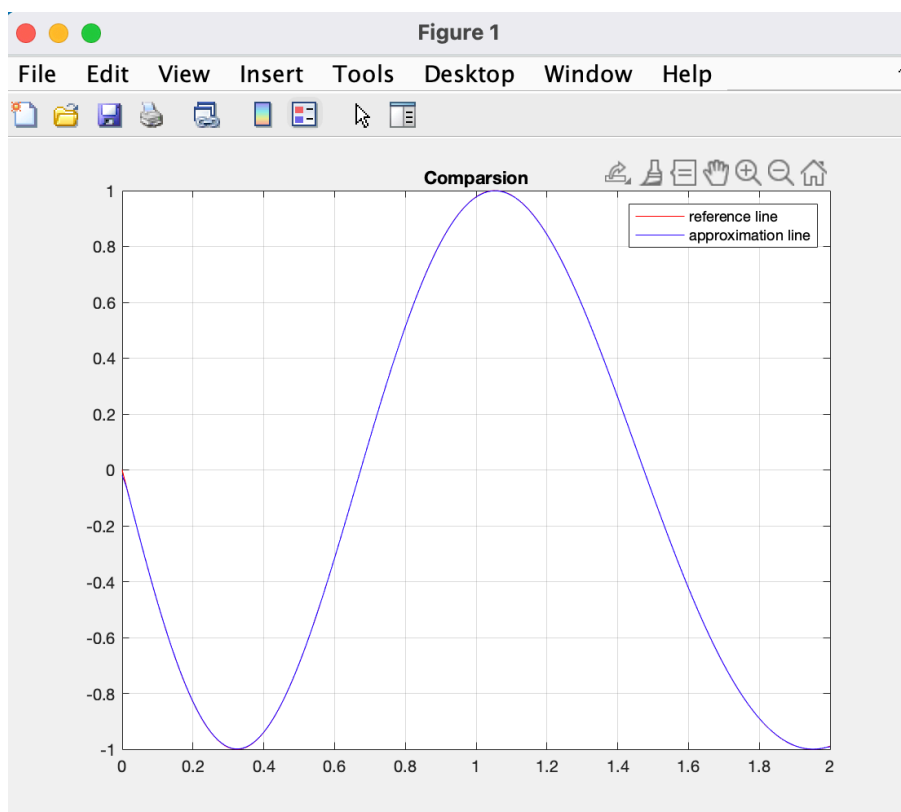
3.

Обучающие подмножество

График функции ошибки



Сравнение эталонных значений с предсказанными

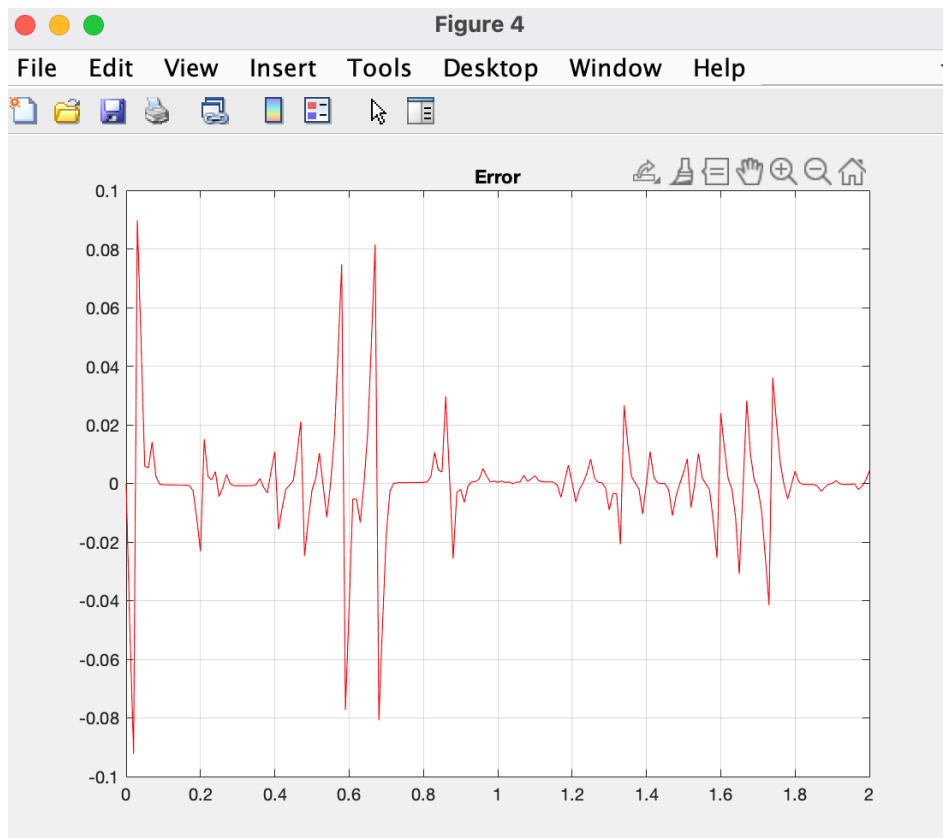




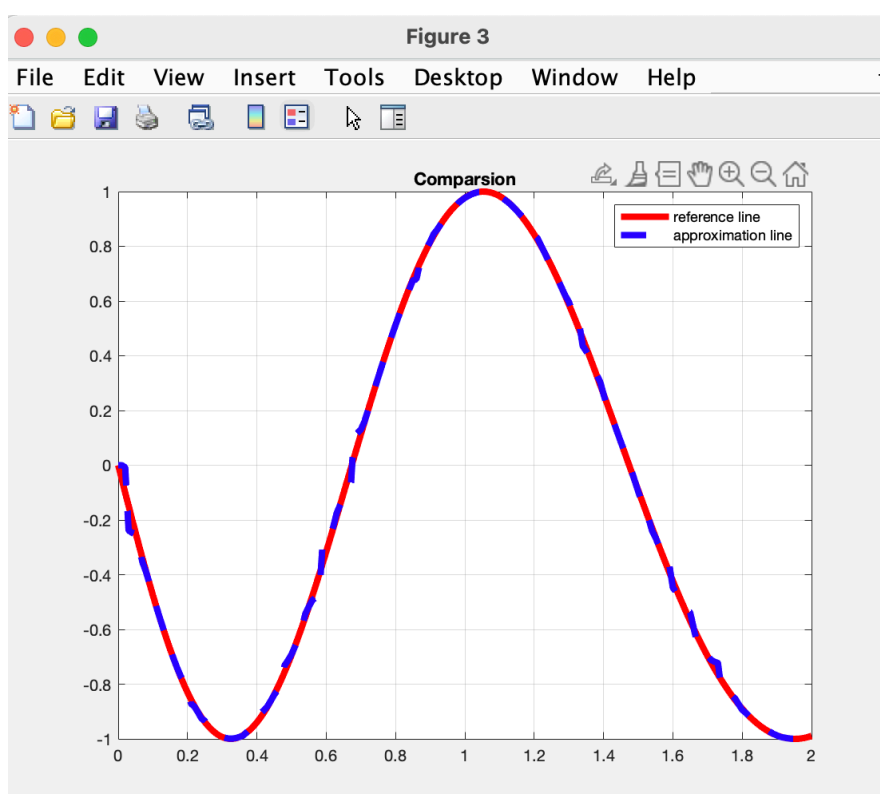
Error on train: 1.757922e-03

Тестовое подмножество

График функции ошибки

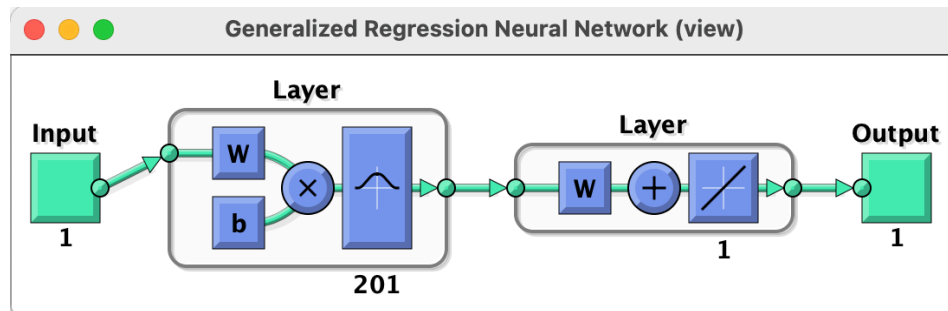


Сравнение эталонных значений с предсказанными



Error on test: 7.335741e-04

## Структура сети



## Выводы

Сети с радиальными базисными элементами можно использовать для классификации линейно неразделимых множеств, регрессии и аппроксимации функции. Во всех случаях малое значение параметра spread давало хорошее качество, хотя в случае малого количества данных, данный подход может приводить к переобучению.

```
% 1.1
% Генерация точек из варианта
% Эллипс: a=0.4, b=0.15,  $\alpha=\pi/6$ ,  $x_0=0.1$ ,  $y_0=-0.15$ 
% Эллипс: a=0.7, b=0.5,  $\alpha=-\pi/3$ ,  $x_0=0$ ,  $y_0=0$ 
% Эллипс: a=1, b=1,  $\alpha=0$ ,  $x_0=0$ ,  $y_0=0$ 
[X1, Y1] = arangePoints(0.4, 0.15, 0.1, -0.15, -pi/6, 0.025);
[X2, Y2] = arangePoints(0.7, 0.5, 0, 0, -pi/3, 0.025);
[X3, Y3] = arangePoints(1, 1, 0, 0, 0, 0.025);

% Разбиение на случайные точки по классам
n = length(X1);
D1 = randperm(n);
n1 = 60;
D1 = D1(1:n1);
K1 = [ones(1, n1); 0*ones(1, n1); 0*ones(1, n1)];

D2 = randperm(n);
n2 = 100;
D2 = D2(1:n2);
K2 = [0*ones(1, n2); ones(1, n2); 0*ones(1, n2)];

D3 = randperm(n);
n3 = 120;
D3 = D3(1:120);
K3 = [0*ones(1, n3); 0*ones(1, n3); ones(1, n3)];

% 1.2
% Разбиение на обучающие и тестовое в соотношении 8 : 2
[trainInd1, testInd1] = dividerand(length(D1), 0.8, 0.2);
[trainInd2, testInd2] = dividerand(length(D2), 0.8, 0.2);
[trainInd3, testInd3] = dividerand(length(D3), 0.8, 0.2);

% 1.3
figure
hold on;

% Отображение первого класса
class1 = plot(X1, Y1, '-r', ...
    'LineWidth', 2);
```

```

tr1 = plot(X1(D1(trainInd1)), Y1(D1(trainInd1)), 'or', ...
'MarkerEdgeColor', 'k', ...
'MarkerFaceColor', 'r', ...
'MarkerSize', 7);

test1 = plot(X1(D1(testInd1)), Y1(D1(testInd1)), 'rs', ...
'MarkerEdgeColor', 'k', ...
'MarkerFaceColor', 'm', ...
'MarkerSize', 7);

% Отображение второго класса
class2 = plot(X2, Y2, '-g', ...
'LineWidth', 2);

tr2 = plot(X2(D2(trainInd2)), Y2(D2(trainInd2)), 'og', ...
'MarkerEdgeColor', 'k', ...
'MarkerFaceColor', 'g', ...
'MarkerSize', 7);

test2 = plot(X2(D2(testInd2)), Y2(D2(testInd2)), 'gs', ...
'MarkerEdgeColor', 'k', ...
'MarkerFaceColor', 'm', ...
'MarkerSize', 7);

% Отображение третьего класса
class3 = plot(X3, Y3, '-b', ...
'LineWidth', 2);

tr3 = plot(X3(D3(trainInd3)), Y3(D3(trainInd3)), 'ob', ...
'MarkerEdgeColor', 'k', ...
'MarkerFaceColor', 'b', ...
'MarkerSize', 7);

test3 = plot(X3(D3(testInd3)), Y3(D3(testInd3)), 'bs', ...
'MarkerEdgeColor', 'k', ...
'MarkerFaceColor', 'm', ...
'MarkerSize', 7);

axis([-1.2 1.2 -1.2 1.2]);
grid on;
title("Classes");

% 1.4
% Объединение в обучающее и тестовое
trainset = [trainInd1, trainInd2+60, trainInd3+160];
testset = [testInd1, testInd2+60, testInd3+160];
X = [X1, X2, X3];
Y = [Y1, Y2, Y3];
D = [D1, D2 + length(X1), D3 + length(X1) + length(X2)];
K = [K1, K2, K3];

% 1.5
% Инициализация сети
spread = 0.3;
net = newpnn([X(D(trainset)); Y(D(trainset))], K(:,trainset), spread);

% 1.7
% Структура сети
view(net);

% 1.8
% Сравнение
PK = net([X(D(trainset)); Y(D(trainset))]);
TK = K(:,trainset);

fprintf('Train size: %d\n', length(trainset));
fprintf('Matches: %d\n\n', sum(TK(TK == PK)));

PK = net([X(D(testset)); Y(D(testset))]);
TK = K(:,testset);

fprintf('Test size: %d\n', length(testset));
fprintf('Matches: %d\n\n', sum(TK(TK == PK)));

% 1.10
% Классификация точек в области
[gX, gY] = meshgrid(-1.2:0.025:1.2, 1.2:-0.025:-1.2);

A = net([gX(:)';gY(:)']);
n = length(gX);
A = max(0, min(1, A));
A = round(A * 10) * 0.1;

ctable = unique(A, 'rows');
cmap = zeros(n, n);

for i = 1 : size(ctable, 1)
    cmap(ismember(A', ctable(i, :), 'rows')) = i;
end

image([-1.2, 1.2], [-1.2, 1.2], cmap);
colormap(ctable);

```

```

% 2.1
% Генерация точек из варианта
% Эллипс: a=0.4, b=0.15,  $\alpha=\pi/6$ , x0=0.1, y0=-0.15
% Эллипс: a=0.7, b=0.5,  $\alpha=-\pi/3$ , x0=0, y0=0
% Эллипс: a=1, b=1,  $\alpha=0$ , x0=0, y0=0
[X1, Y1] = arangePoints(0.4, 0.15, 0.1, -0.15, -pi/6, 0.025);
[X2, Y2] = arangePoints(0.7, 0.5, 0, 0, -pi/3, 0.025);
[X3, Y3] = arangePoints(1, 1, 0, 0, 0, 0.025);

% Разбиение на случайные точки по классам
n = length(X1);
D1 = randperm(n);
n1 = 60;
D1 = D1(1:n1);
K1 = [ones(1, n1); 0*ones(1, n1); 0*ones(1, n1)];

D2 = randperm(n);
n2 = 100;
D2 = D2(1:n2);
K2 = [0*ones(1, n2); ones(1, n2); 0*ones(1, n2)];

D3 = randperm(n);
n3 = 120;
D3 = D3(1:120);
K3 = [0*ones(1, n3); 0*ones(1, n3); ones(1, n3)];

% 2.2
% Разбиение на обучающие и тестовое в соотношении 8 : 2]
[trainInd1, testInd1] = dividerand(length(D1), 0.8, 0.2);
[trainInd2, testInd2] = dividerand(length(D2), 0.8, 0.2);
[trainInd3, testInd3] = dividerand(length(D3), 0.8, 0.2);

% 2.3
figure
hold on;
title("Classes");
% Отображение первого класса
class1 = plot(X1, Y1, '-r', ...
'LineWidth', 2);

tr1 = plot(X1(D1(trainInd1)), Y1(D1(trainInd1)), 'or', ...
'MarkerEdgeColor', 'k', ...
'MarkerFaceColor', 'r', ...
'MarkerSize', 7);

test1 = plot(X1(D1(testInd1)), Y1(D1(testInd1)), 'rs', ...
'MarkerEdgeColor', 'k', ...
'MarkerFaceColor', 'r', ...
'MarkerSize', 7);

% Отображение второго класса
class2 = plot(X2, Y2, '-g', ...
'LineWidth', 2);

tr2 = plot(X2(D2(trainInd2)), Y2(D2(trainInd2)), 'og', ...
'MarkerEdgeColor', 'k', ...
'MarkerFaceColor', 'g', ...
'MarkerSize', 7);

test2 = plot(X2(D2(testInd2)), Y2(D2(testInd2)), 'gs', ...
'MarkerEdgeColor', 'k', ...
'MarkerFaceColor', 'g', ...
'MarkerSize', 7);

% Отображение третьего класса
class3 = plot(X3, Y3, '-b', ...
'LineWidth', 2);

tr3 = plot(X3(D3(trainInd3)), Y3(D3(trainInd3)), 'ob', ...
'MarkerEdgeColor', 'k', ...
'MarkerFaceColor', 'b', ...
'MarkerSize', 7);

test3 = plot(X3(D3(testInd3)), Y3(D3(testInd3)), 'bs', ...
'MarkerEdgeColor', 'k', ...
'MarkerFaceColor', 'b', ...
'MarkerSize', 7);

legend('Set 1', 'Train set 1', 'Test set 1',...
'Set 2', 'Train set 2', 'Test set 2',...
'Set 3', 'Train set 3', 'Test set 3');

```

```
% 2.4
% Объединение в обучающее и тестовое
trainset = [trainInd1, trainInd2+60, trainInd3+160];
testset = [testInd1, testInd2+60, testInd3+160];
X = [X1, X2, X3];
Y = [Y1, Y2, Y3];
D = [D1, D2 + length(X1), D3 + length(X1) + length(X2)];
K = [K1, K2, K3];

% 2.5
% Инициализация сети
spread = 0.3;
goal = 1.0e-5;
net = newrb([X(D(trainset)); Y(D(trainset))], K(:,trainset), goal, spread);

% 2.6
% Структура сети
view(net);

% 2.8
% Сравнение
PK = net([X(D(trainset)); Y(D(trainset))]);
TK = K(:,trainset);

fprintf('Train size: %d\n', length(trainset));
fprintf('Matches: %d\n', sum(TK(TK == PK)));

PK = net([X(D(testset)); Y(D(testset))]);
TK = K(:,testset);

fprintf('Test size: %d\n', length(testset));
fprintf('Matches: %d\n', sum(TK(TK == PK)));

% 2.9
% Классификация точек в области
figure
hold on;
axis([-1.2 1.2 -1.2 1.2]);
grid on

[gX, gY] = meshgrid(-1.2:0.025:1.2, 1.2:-0.025:-1.2);
```

```
A = net([gX(:)';gY(:)']);
n = length(gX);
A = max(0, min(1, A));
A = round(A * 10) * 0.1;

ctable = unique(A', 'rows');
cmap = zeros(n, n);

for i = 1 : size(ctable, 1)
    cmap(ismember(A', ctable(i, :), 'rows')) = i;
end
image([-1.2, 1.2], [-1.2, 1.2], cmap);
colormap(ctable);

hold on;
axis([-1.2 1.2 -1.2 1.2]);
title("Points in area");
grid on

% 2.10
spread = 0.1;
goal = 1.0e-5;
net = newrb([X(D(trainset)); Y(D(trainset))], K(:,trainset), goal, spread);
view(net);

% 2.11
% Классификация точек в области
A = net([gX(:)';gY(:)']);
n = length(gX);
A = max(0, min(1, A));
A = round(A * 10) * 0.1;

ctable = unique(A', 'rows');
cmap = zeros(n, n);

for i = 1 : size(ctable, 1)
    cmap(ismember(A', ctable(i, :), 'rows')) = i;
end
image([-1.2, 1.2], [-1.2, 1.2], cmap);
colormap(ctable);
```

```
% 3.1
% Использование обобщенно-регрессионной нейронной сети
t0 = 0;
tn = 2;
h = 0.01;
n = (tn - t0) / h + 1;

t = t0:h:tn;
x = sin(0.5 * (t.^2) - 5 * t);

% 3.2
% Инициализация сети
spread = h;
[trainInd, testInd] = dividerand(n, .8, .2);
net = newgrnn(t, x, spread);
view(net);

y = sim(net, t);
fprintf('Error on train: %d\n', sqrt(mse(x(trainInd) - y(trainInd))));
fprintf('Error on test: %d\n', sqrt(mse(x(testInd) - y(testInd))));

% 3.5
figure
referenceLine = plot(t, x, 'r');

hold on;
approximationLine = plot(t, y, '-b');

grid on;
legend([referenceLine,approximationLine],'reference line', 'approximation line');
title("Comparsion");

figure;
plot(t, x - y, 'r');
title("Error");
grid on

net = newgrnn(t(trainInd), x(trainInd), spread);
y = sim(net, t);
figure
referenceLine = plot(t, x, 'r');

hold on;
approximationLine = plot(t, y, '--b');
set(approximationLine, 'linewidth', 4);

grid on;
legend([referenceLine,approximationLine],'reference line', 'approximation line');
title("Comparsion");

figure;
plot(t, x - y, 'r');
title("Error");
grid on;
```