

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

Лабораторная работа № 8
по курсу «Нейроинформатика»
Тема: Динамические сети.

Студент: Ваньков Д. А.
Группа: 8О-407Б-17
Преподаватель: Аносова Н.П.

Москва, 2021

Постановка задачи

Исследование свойств некоторых динамических нейронных сетей, алгоритмов обучения, а также применение сетей в задачах аппроксимации функций и распознавания динамических образов.

1. Использовать сеть прямого распространения с запаздыванием для предсказания значений временного ряда и выполнения многошагового прогноза.
2. Использовать сеть прямого распространения с распределенным запаздыванием для распознавания динамических образов.
3. Использовать нелинейную авторегрессионную сеть с внешними входами для аппроксимации траектории динамической системы и выполнения многошагового прогноза.

Вариант 6:

05/1811

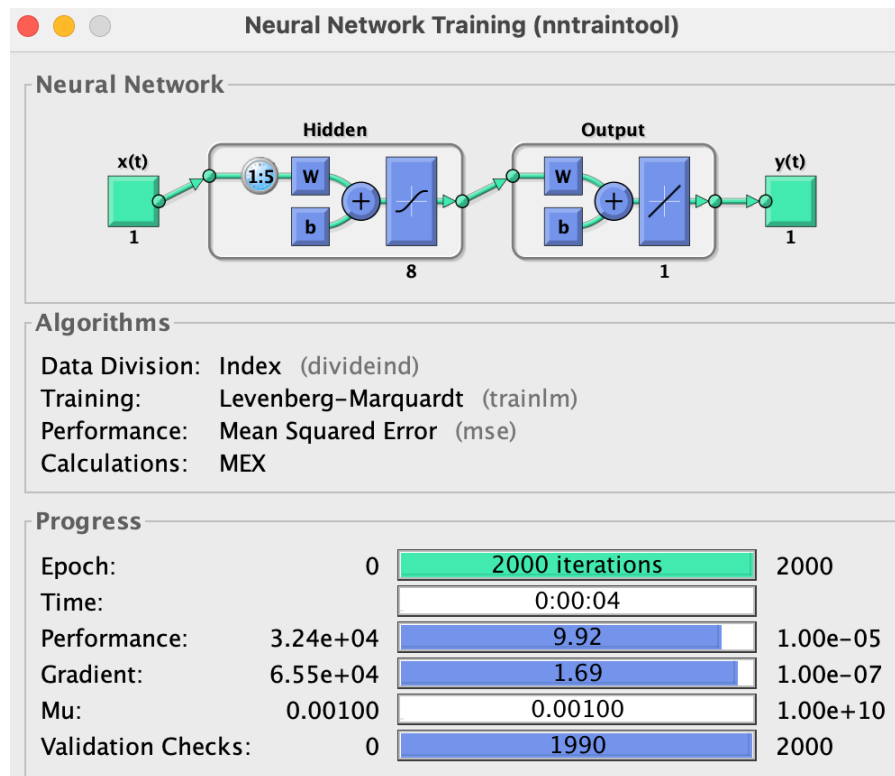
$$6. \quad \left| \begin{array}{l} u(k) = \sin(k^2 - 6k + 3) \end{array} \right.$$

$$\left| g(k) = \cos(\cos(k)k^2 + 5k), \quad k \in [2.38, 4.1] \right| \quad [1, 3, 5]$$

Ход работы

1 (Аппроксимация чисел Вольфа)

Структура сети



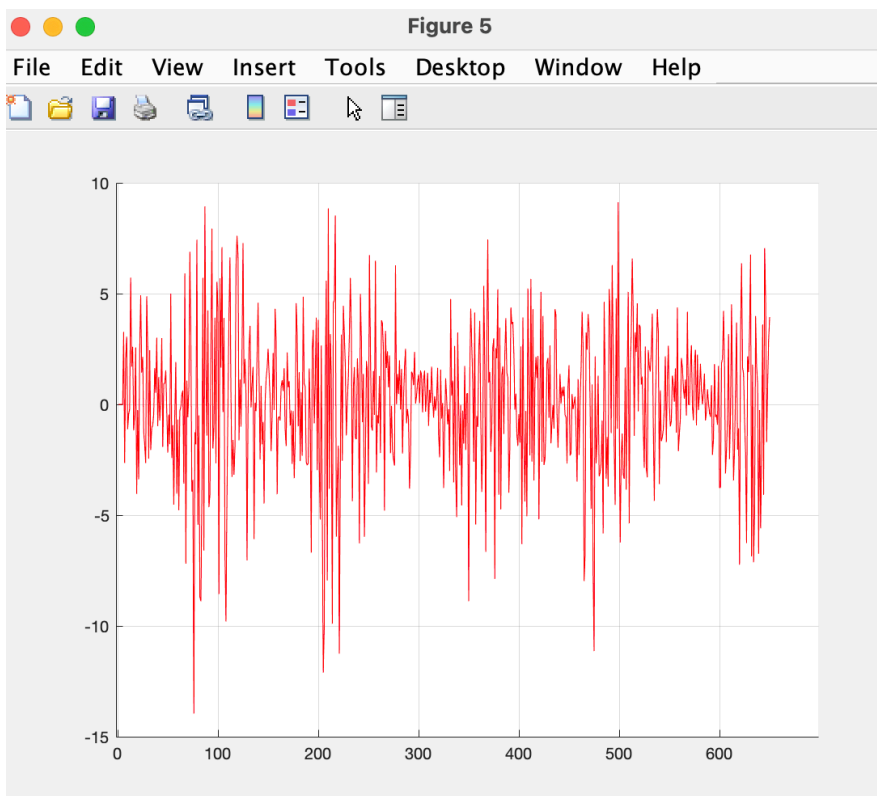
Выход сети для обучающего подмножества



Выход сети для тестового подмножества

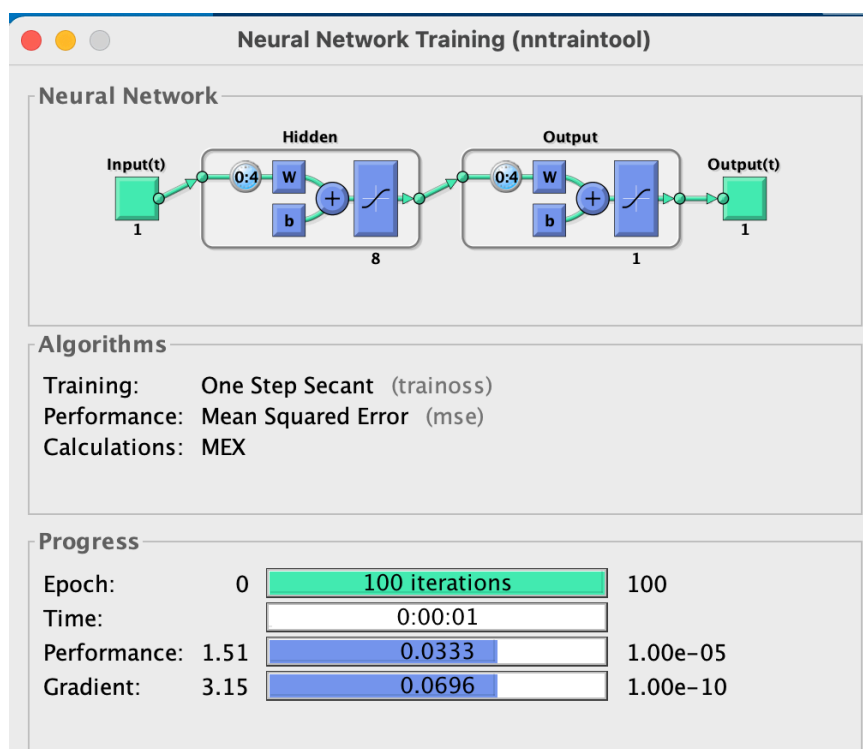
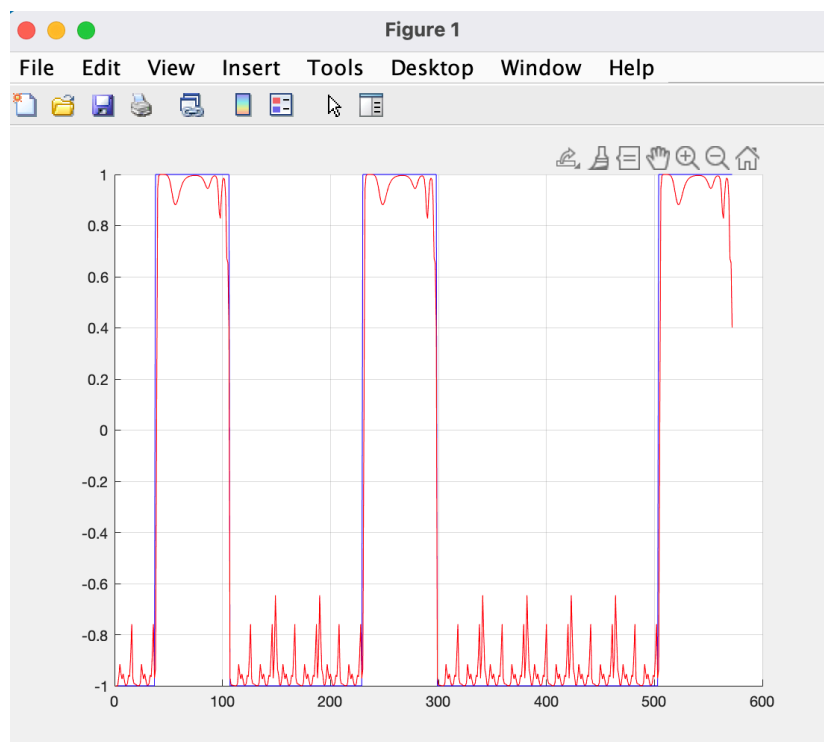


График ошибки

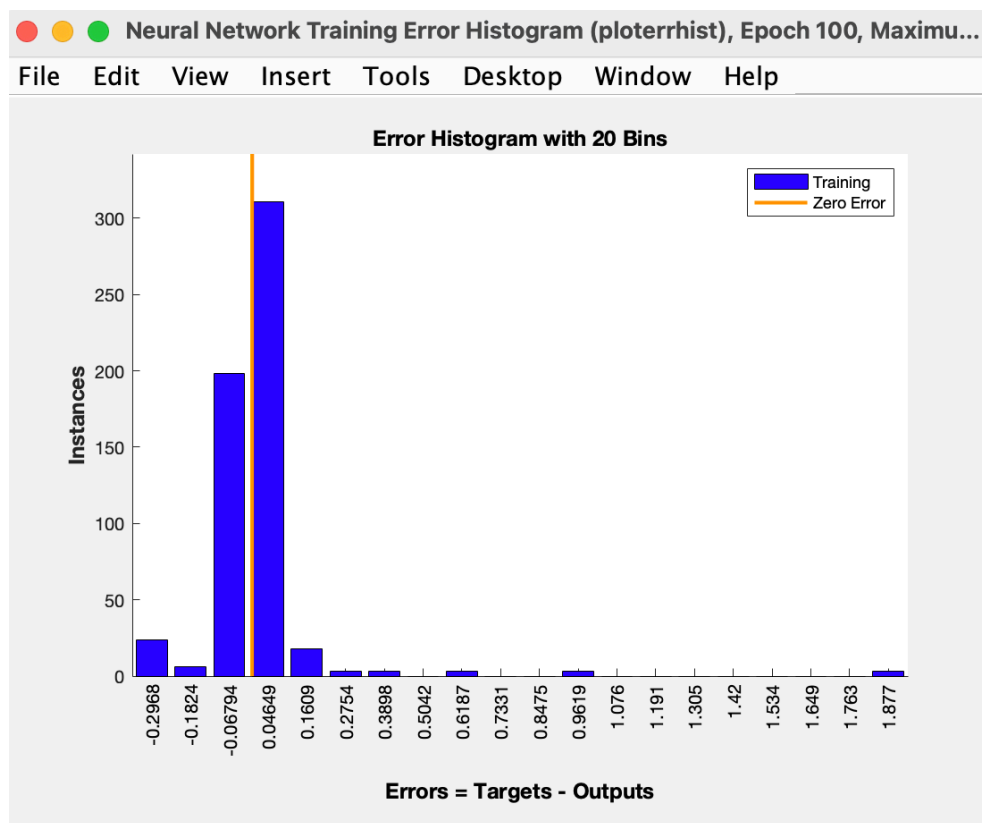


2 (Распознавание динамических образов)

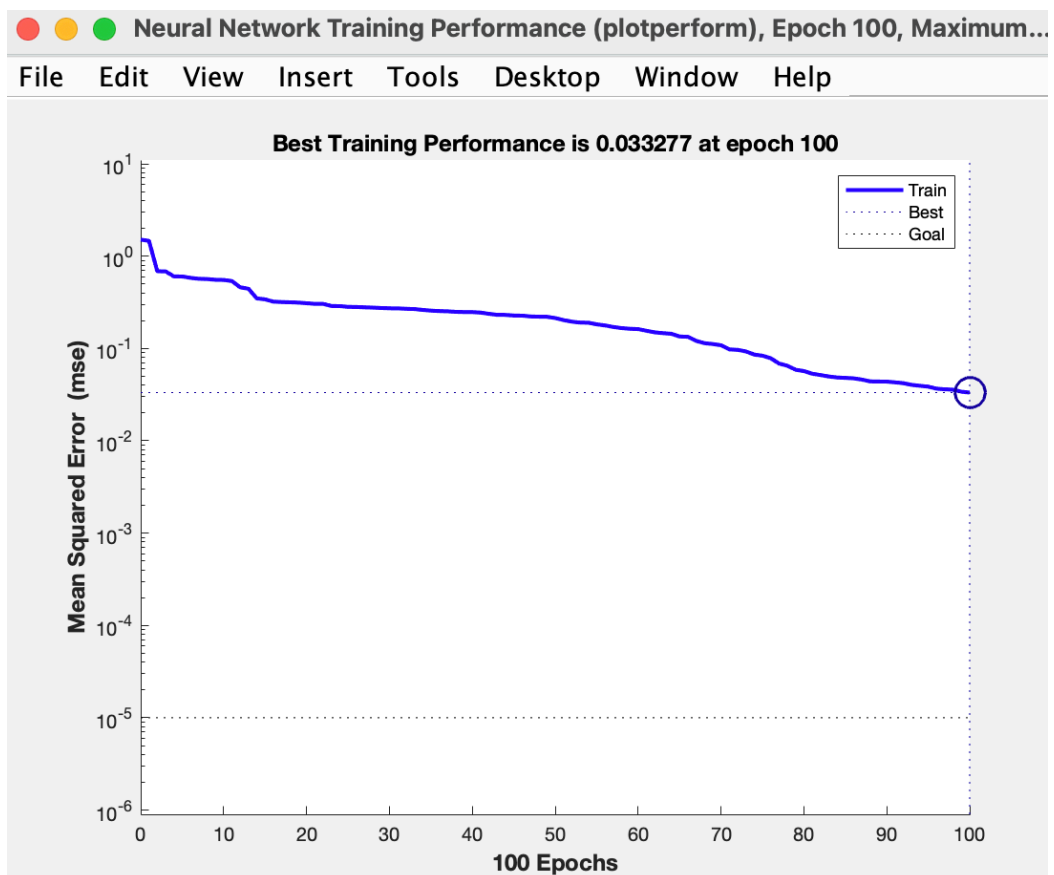
Выход сети для обучающего множества



Error

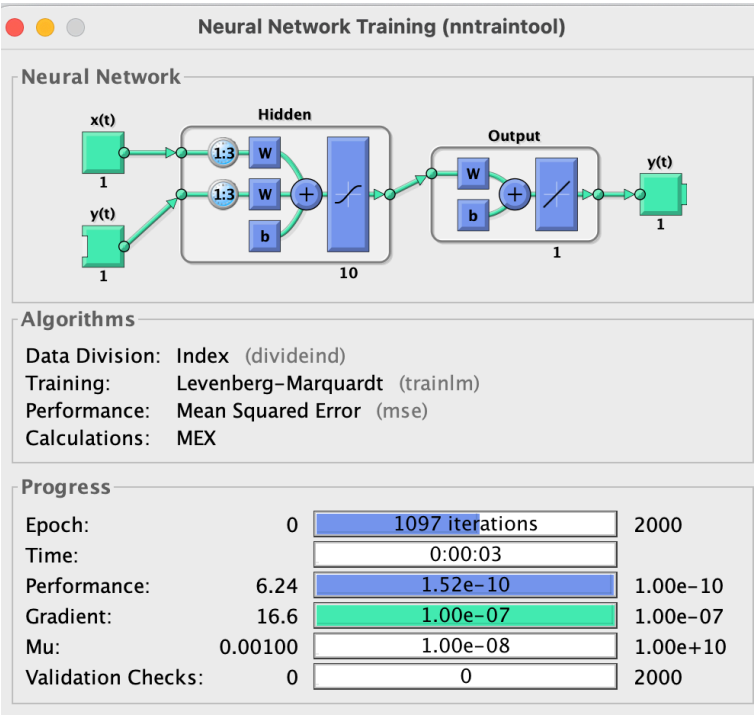


Performance

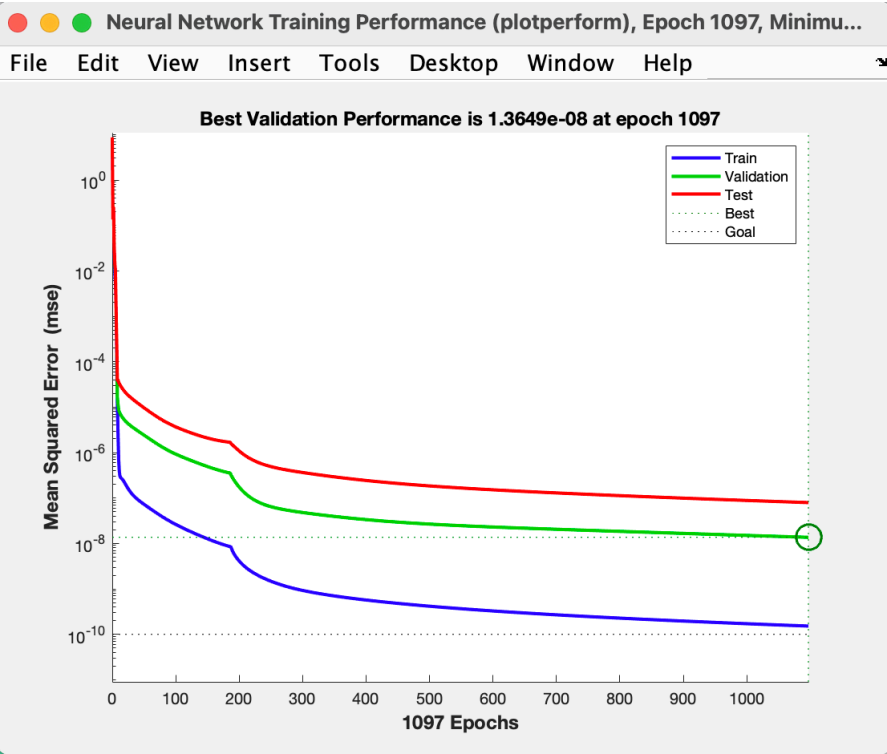


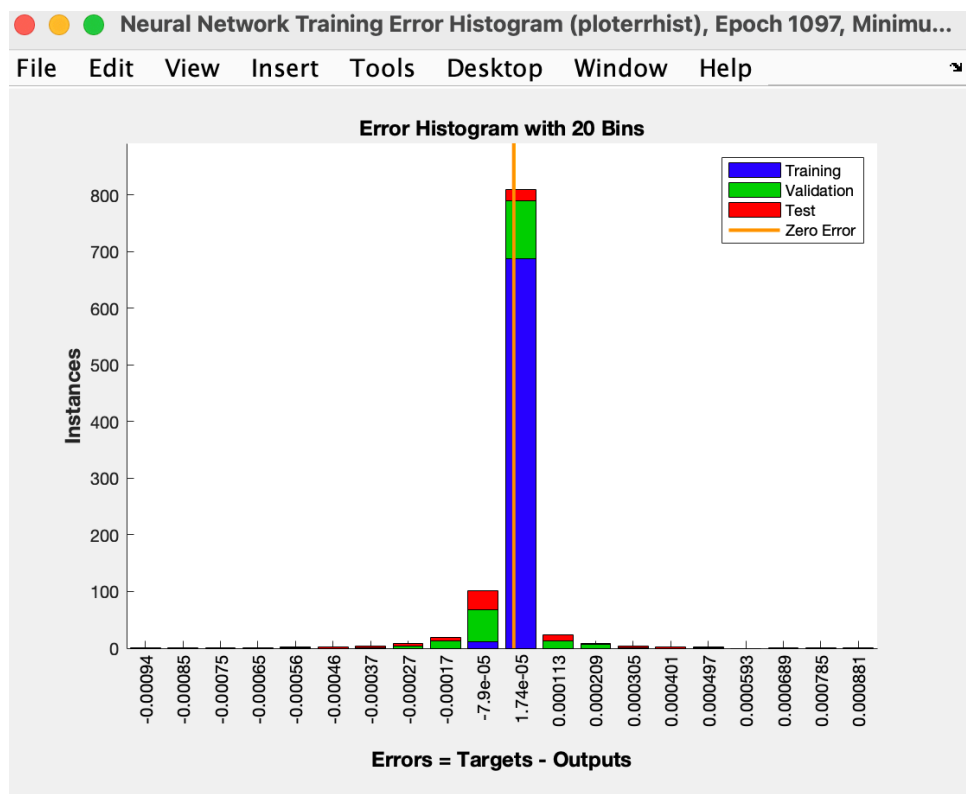
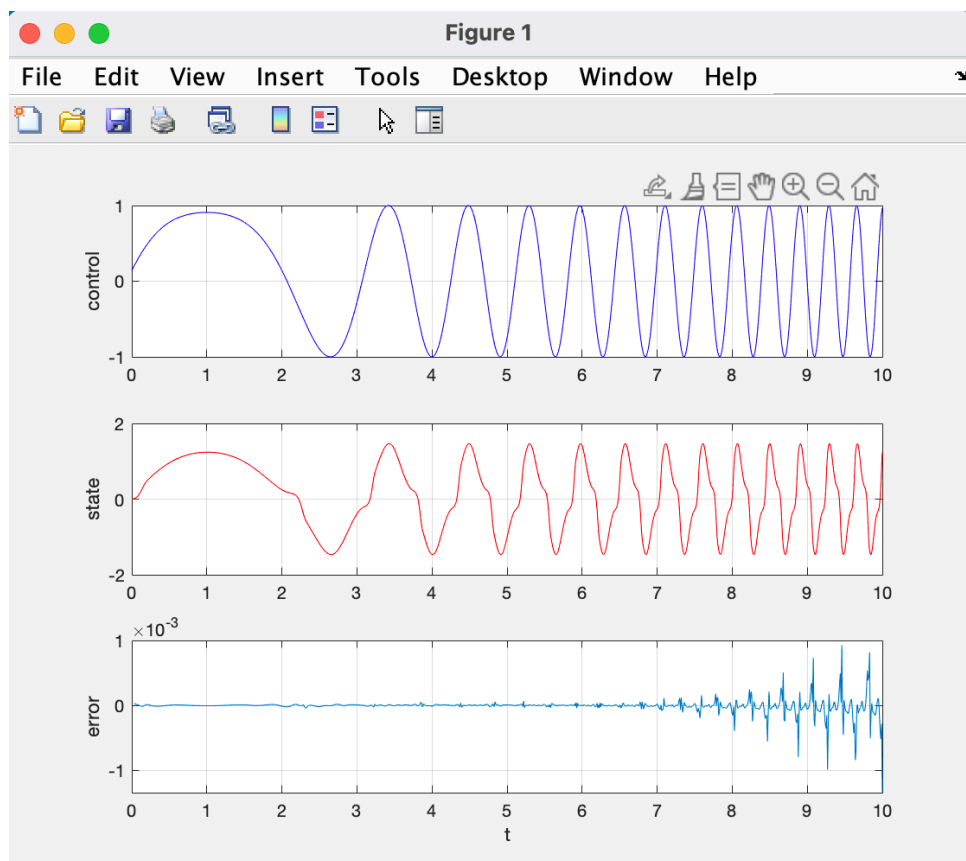
3 (Аппроксимация траектории)

Структура сети



Performance






```

% 1.1 - 1.2
% Парсим данные
fileID = fopen('Data.txt','r');
formatSpec = '%f %f';
sizeData = [1 Inf];
wolf_dataset = fscanf(fileID,formatSpec,sizeData);
fclose(fileID);

% 1.3
% Сглаживание
x = wolf_dataset;
x = smooth(x, 12);

% 1.4
% Задаем глубину погружения временного ряда
D = 5;
ntrain = 500;
nval = 100;
ntest = 50;

% 1.5
% Объединяем подмножества в обучающую выборку
% 1 - 500
trainInd = 1 : ntrain;
% 501 - 600
valInd = ntrain + 1 : ntrain + nval;
% 601 - 650
testInd = ntrain + nval + 1 : ntrain + nval + ntest;

% 1.7
% Создаем сеть
net = timedelaynet(1:D,8,'trainlm');
net.divideFcn = 'divideind';
net.divideParam.trainInd = trainInd;
net.divideParam.valInd = valInd;
net.divideParam.testInd = testInd;

```

```

% 1.6
% Преобразуем обучающее множество
x = con2seq(x(1:ntrain+nval+ntest));

% 1.9
% Конфигурируем сеть под обучающее множество
net = configure(net, x, x);

% 1.10
% Инициализируем весовые коэффициенты
net = init(net);

% 1.11
% Задаем параметры обучения
net.trainParam.epochs = 2000;
net.trainParam.max_fail = 2000;
net.trainParam.goal = 1.0e-5;
view(net);

% 1.12
% Обучаем сеть
[Xs, Xi, Ai, Ts] = preparets(net, x, x);
net = train(net, Xs, Ts, Xi, Ai);

% 1.14
% Расчитываем выход сети
Y = sim(net, Xs, Xi);

figure;
hold on;
grid on;
plot(cell2mat(x), '-b');
plot([cell2mat(Xi) cell2mat(Y)], '-r');

figure;
hold on;
grid on;
plot([cell2mat(Xi) cell2mat(Y)] - cell2mat(x), '-r');

xm = cell2mat(x);
ym = cell2mat(Y);

```

```

figure;
hold on;
grid on;
plot(xm(ntrain + nval + 1 : ntrain + nval + ntest), '-b');
plot(ym(ntrain + nval - 9 : ntrain + nval + ntest - 10), '-r');|

```

```

% 2.1
% Создаем обучающее множество
k1 = 0 : 0.025 : 1;
p1 = sin(4 * pi * k1);
t1 = -ones(size(p1));
k2 = 2.38 : 0.025 : 4.1;
g = @(k)cos(cos(k) .* k.^2 + 5*k);
p2 = g(k2);
t2 = ones(size(p2));
R = {1; 3; 5};
P = [repmat(p1, 1, R{1}), p2, repmat(p1, 1, R{2}), p2, repmat(p1, 1, R{3}), p2];
T = [repmat(t1, 1, R{1}), t2, repmat(t1, 1, R{2}), t2, repmat(t1, 1, R{3}), t2];
Pseq = con2seq(P);
Tseq = con2seq(T);

% 2.2
% Создаем сеть
net = distdelaynet([0 : 4, 0 : 4], 8, 'trainoss');
net.layers{1}.transferFcn = 'tansig';
net.layers{2}.transferFcn = 'tansig';
net.divideFcn = '';

% 2.3
% Конфигурируем сеть
net = configure(net, Pseq, Tseq);
view(net);

% 2.4
% Формируем массивы ячеек для функции обучения
[Xs, Xi, Ai, Ts] = preparets(net, Pseq, Tseq);

% 2.5
% Задаем параметры обучения
net.trainParam.epochs = 100;
net.trainParam.goal = 1.0e-5;

```

```

% 2.6
% Обучаем сеть
net = train(net, Xs, Ts, Xi, Ai);

% 2.8
% Расчитываем выход сети
Y = sim(net, Xs, Xi, Ai);

figure;
hold on;
grid on;
plot(cell2mat(Ts), '-b');
plot(cell2mat(Y), '-r');

% 2.9
% Преобразовываем значения по правилу
Yc = zeros(1, numel(Y));
for i = 1 : numel(Y)
    if Y{i} >= 0
        Yc(i) = 1;
    else
        Yc(i) = -1;
    end
end

```

```

% 3.1
% Создаем обучающее множество
t0 = 0;
tn = 10;
dt = 0.01;
n = (tn - t0) / dt + 1;
fun = @(k)sin(k.^2 - 2 * k + 3);
fun2 = @(y, u) y ./ (1 + y.^2) + u.^3;
u = zeros(1, n);
u(1) = fun(0);
x = zeros(1, n);
for i = 2 : n
    t = t0 + (i - 1) * dt;
    x(i) = fun2(x(i - 1), u(i - 1));
    u(i) = fun(t);
end

% 3.2
% Задаем глубину погружения временного ряда
D = 3;
ntrain = 700;
nval = 200;
ntest = 97;

% 3.3
% Объединяем подмножества в обучающую выборку
trainInd = 1 : ntrain;
valInd = ntrain + 1 : ntrain + nval;
testInd = ntrain + nval + 1 : ntrain + nval + ntest;

% 3.5
% Создаем сеть
net = narxnet(1 : 3, 1 : 3, 10);
net.trainFcn = 'trainlm';

% 3.6
% Разделяем обучающее множество
net.divideFcn = 'divideind';
net.divideParam.trainInd = trainInd;
net.divideParam.valInd = valInd;
net.divideParam.testInd = testInd;

```

```

% 3.9
% Задаем параметры обучения
net.trainParam.epochs = 2000;
net.trainParam.max_fail = 2000;
net.trainParam.goal = 1.0e-10;

[Xs, Xi, Ai, Ts] = preparets(net, con2seq(u), {}, con2seq(x));

% 3.10
% Обучаем сеть
net = train(net, Xs, Ts, Xi, Ai);
view(net);

% 3.12
% Рассчитываем выход сети
Y = sim(net, Xs, Xi);

figure
subplot(3,1,1)
plot(t0:dt:tn, u, '-b'), grid
ylabel('control')
subplot(3,1,2)
plot(t0:dt:tn, x, '-b', t0:dt:tn, [x(1:D) cell2mat(Y)], '-r'), grid
ylabel('state')
subplot(3,1,3)
plot(t0+D*dt:dt:tn, x(D+1:end) - cell2mat(Y)), grid
ylabel('error')
xlabel('t')

```

Вывод

Такие сети используются в качестве ассоциативной памяти, а также в задачах нелинейной фильтрации данных, для предсказания и распознавания закономерностей.

Выполнив данную лабораторную я научился применять динамические сети для предсказания значений временного ряда и выполнения многошагового прогноза, распознавания динамических образов и аппроксимации траектории динамической системы.